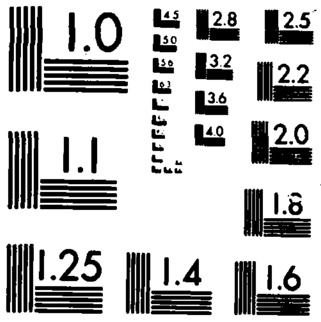
AD-A164 257 THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. C A BAXLEY 1/5

UNCLASSIFIED DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 NL


1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

1

AD-A164 257

[illegible]
MODEL OF THE MOTOROLA MC68000
MICROPROCESSOR WITH N.MPC.

THESIS
(2 of 3) Volum

Charles A. Baxley Jr.
Captain, USAF

AFIT/GCS/ENG/84D-2 -Vol-2

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 12 025

1

AFIT/GCS/ENG/84D-2 -Vol-2

THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROPROCESSOR WITH N.MPC.

THESIS
(2 of 3)

Volume 2
appendices A-G.

Charles A. Baxley Jr.
Captain, USAF

AFIT/GCS/ENG/84D-2 -Vol-2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A164 257

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GCS/ENG/84-D-2	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering — 6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFSC/FTD — 8b. OFFICE SYMBOL (If applicable) TQTA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	

12. PERSONAL AUTHOR(S)
Charles A. Baxley Jr., B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1984 December	1200

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Computer Design Language(CDL), Instruction Set Processor (ISP'), N.mPc(Networked Microprocessor) Motorola MC68000, Microprocessor Modeling,
09	02		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROPROCESSOR WITH N.mPc

Thesis Chairman: Frederick A. Zapka, Major, USA

Approved for public release: IAW AFR 190-1.
[illegible] 14 AUG. 85
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AFIT)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED [✓] SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Frederick A. Zapka, Major, USA	22b. TELEPHONE NUMBER (Include Area Code) [illegible]	22c. OFFICE SYMBOL AFIT/ENG

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION

18. Microprocessor Simulation, Microprocessor Analysis, Computer Architecture, Microcomputers, Computerized Simulation, Digital Simulation.

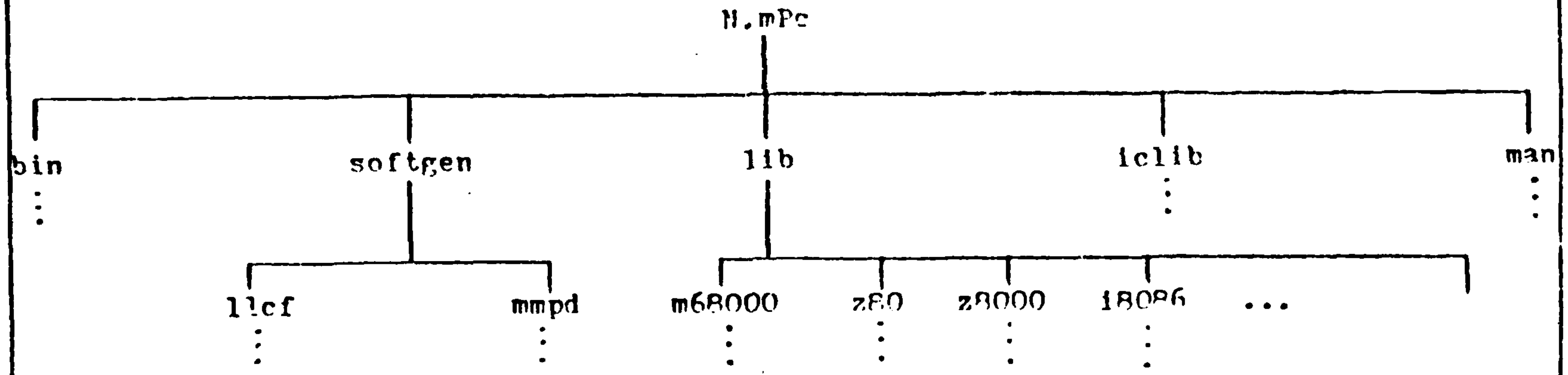
19. In a prior thesis project, a functional level model of portions of the Motorola MC68000 microprocessor was developed using signal analysis supported by limited technical data. Representative parts of the instruction set and exception processing structure were modeled with the Computer Design Language (CDL). In this follow-on effort, those CDL models are transformed into equivalent models using ISP', an enhanced version of the Instruction Set Processor (ISP) hardware design language. This language transformation enabled the models to be simulated using N.mPc, a VAX 11/780-hosted software package developed specifically to support the design of digital systems. To evaluate the correctness of the of the models, the simulation results are analyzed against signal data gathered with the aid of a logic analyzer during the actual operation of the MC68000 when processing the modeled instructions. The accuracy and completeness of the examined models suggests that this functional approach to microprocessor modeling is a valid one.

Contents

(Volume I)

Appendix A: Local N.mPc Supplement

This appendix provides information on the access and use of N.mPc as locally installed on AFIT's SCC VAX 11/780. Included is:



Note: To use N.mPc, install the path "/usr/nmpc/bin" in your .cshrc file.

Note: The directory "/usr/nmpc/lib" houses N.mPc's microprocessor library for constructing simulations. Its subdirectories contain numerous examples of isp', Metamicro, and Linking/Loader descriptions.

Local N.mPc Directory Structure

nmpc/lib

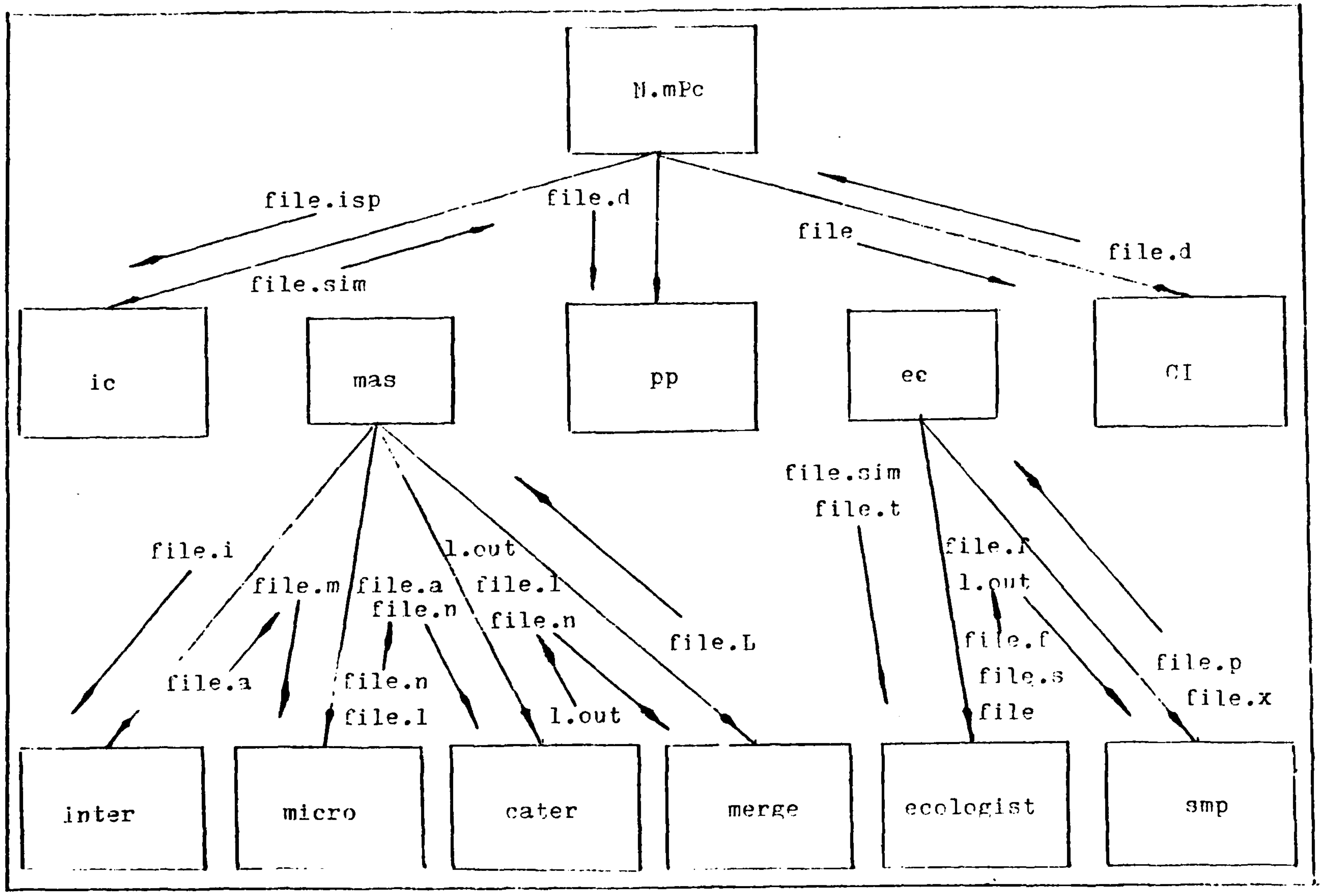
```
./lib:
./
../
a2900/
bwald1/
bwald2/
i8048/
i8051/
i8080/
i8085/
i8086/
m6502/
m6800/
m68000/
nebula/
pdp11/
r1802/
s2650/
t9900/
t1320/
vax11/
z80/
z8000/

./lib/a2900:
./
../
RCS/
a2901a.lsp
a2902a.lsp
a2909a.lsp
a2910a.lsp
a2911a.lsp

./lib/a2900/RCS:
./
../
a2901a.lsp,v
a2902a.lsp,v
a2909a.lsp,v
a2910a.lsp,v
a2911a.lsp,v

./lib/bwald1:
./
../
RCS/
bwald1.a
bwald1.l
bwald1.m

./lib/bwald1/RCS:
./
../
bwald1.a,v
bwald1.l,v
bwald1.m,v

./lib/bwald2:
./
```

```
nebulaa.lsps
nebulaaf.lsps
nebulaal.lsps

./lib/nebula/RCS:
./
../
float.lsps,v
lex.lsps,v
nebula.lsps,v
nebulaa.lsps,v

./lib/pdp11:
./
../
RCS/
pdp11.l
pdp11.m

./lib/pdp11/RCS:
./
../
pdp11.l,v
pdp11.m,v

./lib/r1802:
./
../
RCS/
doc/
r1802.l
r1802.m
r1802a.lsp
r1802a.sim
r1802c.lsp
r1802cm.lsp

./lib/r1802/RCS:
./
../
r1802.l,v
r1802.m,v
r1802a.lsp,v
r1802a.sim,v
r1802c.lsp,v
r1802cm.lsp,v

./lib/r1802/doc:
./
../
cosmac/

./lib/r1802/doc/cosmac:
./
../
makeum
r1802
r1802.appx
r1802.ex1
r1802.ex2
r1802.fd
r1802.l
r1802.tc
r1802.tp
```

```
r1802a.lsp
r1802b.lsp
r1802m.lsp

./lib/s2650:
./
../
RCS/
s2650.l
s2650.m
s2650a.lsp

./lib/s2650/RCS:
./
../
s2650.l,v
s2650.m,v
s2650a.lsp,v

./lib/t9900:
./
../
RCS/
t9900.l
t9900.m
t9900a.lsp
t9900b.lsp

./lib/t9900/RCS:
./
../
t9900.l,v
t9900.m,v
t9900a.lsp,v
t9900b.lsp,v

./lib/t1320:
./
../
RCS/
READ_ME
ti.lsp
ti.read
tid.a
tid.l
tid.m
tip.a
tip.l
tip.m

./lib/t1320/RCS:
./
../
ti.lsp,v
ti.read,v
tid.l,v
tid.m,v
tip.l,v
tip.m,v

./lib/vax11:
./
../
RCS/
```

```
../
IObus.isp
RCS/
bwald2.a
bwald2.i
bwald2.isp
bwald2.m
bwald2.t
clock.isp
mem.isp

./lib/bwald2/RCS:
./
../
IObus.isp,v
bwald2.a,v
bwald2.i,v
bwald2.isp,v
bwald2.m,v
bwald2.t,v
clock.isp,v
mem.isp,v

./lib/i8048:
./
../
RCS/
i8048a.i
i8048a.isp
i8048a.m

./lib/i8048/RCS:
./
../
i8048a.i,v
i8048a.isp,v
i8048a.m,v

./lib/i8051:
./
../
RCS/
datam.isp
i8051.i
i8051.isp
i8051.m
i8051.t
progm.isp
test

./lib/i8051/RCS:
./
../
datam.isp,v
i8051.i,v
i8051.isp,v
i8051.m,v
i8051.t,v
progm.isp,v

./lib/i8080:
./
../
RCS/
i8080.i
i8080.m
i8080e.isp

./lib/i8080/RCS:
./
../
i8080.i,v
i8080.m,v
i8080e.isp,v

./lib/i8085:
./
../
RCS/
i8085.i
i8085.m
i8085a.isp
i8085b.isp

./lib/i8085/RCS:
./
../
i8085.i,v
i8085.m,v
i8085a.isp,v
i8085b.isp,v

./lib/i8086:
./
../
8086.a
8086.i
8086.m
8086.n
RCS/
e8086max.isp
e8086max.sim
e8086min.isp
e8086min.sim
i8086max.isp
i8086max.sim
i8086min.isp
i8086min.sim
min86mem.isp
min86mem.sim
read_me

./lib/i8086/RCS:
./
../
8086.i,v
8086.m,v
e8086max.isp,v
e8086min.isp,v
i8086max.isp,v
i8086min.isp,v
min86mem.isp,v

./lib/m6502:
./
../
RCS/
m6502.i
m6502.isp
m6502.m

./lib/m6502/RCS:
./
../
m6502.i,v
m6502.isp,v
m6502.m,v

./lib/m6800:
./
../
RCS/
m6800.i
m6800.m

./lib/m6800/RCS:
./
../
m6800.i,v
m6800.m,v

./lib/m68000:
./
../
68000.a
68000.i
68000.m
68000.n
RCS/
e68000.isp
e68000.sim
e68000m.isp
e68000m.sim
i68000.isp
i68000.sim
m68000.a
m68000.i
m68000.m
m68000a.isp
m68000a.sim
m68000b.isp
m68000b.sim
m68000bm.isp
m68000bm.sim
read_me

./lib/m68000/RCS:
./
../
68000.i,v
68000.m,v
e68000.isp,v
e68000m.isp,v
i68000.isp,v

./lib/nebula:
./
../
RCS/
float.isps
lex.isps
nebula.isps
```

```
vax11a.lsp

./lib/vax11/RCS:
./
../
vax11a.lsp,v

./lib/z80:
./
../
RCS/
z80.doc
z80.l
z80.m
z80a.lsp
z80b.lsp
z80c.lsp
z80d.lsp
z80e.lsp

./lib/z80/RCS:
./
../
z80.doc,v
z80.l,v
z80.m,v
z80a.lsp,v
z80b.lsp,v
z80c.lsp,v
z80d.lsp,v
z80e.lsp,v

./lib/z8000:
./
../
RCS/
z8000.l
z8000.m
z8000a.lsp
z8000b.lsp
z8000bm.lsp

./lib/z8000/RCS:
./
../
z8000.l,v
z8000.m,v
z8000a.lsp,v
z8000b.lsp,v
z8000bm.lsp,v
```



N.mPc Organization

File Names

Many different file names are involved in a simulation, and it was considered desirable to summerize them in one place.

xxxx.m	A ".m" file is the source input file to the metaMicro assembler. A successful assembly produces a corresponding "xxxx.n" file.
xxxx.n	The intermediate file produced by the metaMicro assembler. Used by the Linking/Loader Allocator.
yyyy.i	The source input to the Linking/Loader Interpreter. Contains the specification of the address resolution process for a given machine.
yyyy.a	The output of the Linking/Loader Interpreter. This file is used by the Allocator to direct the address resolution process.
l.out	The output of the Allocator. Contains a real machine core image, suitable for simulation after processing by the simulated memory processor.
zzzz.isp	The input to the ISP' compiler, contains ISP' source code.
zzzz.sim	The output of the ISP' compiler, corresponds to the zzzz.isp source input.
root.t	The topology file, the ecologist will build a program called "root", which will be the executable simulation.
root.s	A symbol table file created by the ecologist, used by the runtime package. "root" is the simulation name.
root.f	The memory list file, produced by the ecologist, and used by the simulated memory processor. Contains the names of all memories used in a simulation.
wwww.p	A processed file created by "smp". Corresponds to a previous Linking/Loader output file which has been renamed. Used by the simulation program and the Simulated Memory Editor. One for each non raw

memory in a simulation.

root.x	Another output of the smp. Contains the global symbols passed on from metaMicro. One per simulation.
root.d	Simulation data file, produced by the simulation itself, contains data to be processed by the N.mPc post processor.

NAME

cater - Linking/Loader Allocator

SYNOPSIS

cater [-options] file.a file1.n [file2.n ... file10.n]

DESCRIPTION

Cater is the second program of the Linking/Loader. Cater is driven by a user description of the address resolution process produced by the Interpreter.

Up to 10 files produced by metaMicro may be linked at one time.

The description file produced by the Interpreter has a ".a" extension and the metaMicro output files have a ".n" extension.

The Allocator produces a file with the name l.out. This file can be merged with metaMicro listing files with the merge program. A dump of this file may be obtained from the 'mdump' program.

OPTIONS

L	print the "Logical Space Map", relating logical addresses (produced with the -p option of micro) to physical addresses as established by cater.
P	print the "Physical Space Map", relating physical addresses to logical addresses. The "Physical Space Map" is the inverse of the "Logical Space Map".
E	print the "External Symbol Table", relating external symbols identified by the global label declaration in metaMicro to physical addresses where allocated by cater.
F	print the "Free Space After Allocation" report. This report shows the unused blocks of the target machine memory space as defined by the space declaration in the Linking/Loader command program.
T	print the "Transfers Located" report. This report shows where transfers have been placed into the target machine code to maintain logical contiguity of physically disjoint segments of target machine code.
R	print the "Allocation Order" report, showing the order in which target machine instructions were allocated.
A	print all reports noted above.
H	print all physical addresses and complete instructions in hexidecimal.
D	print all physical addresses in decimal. Complete instructions are printed in octal.
O	print all physical addresses and complete instructions in octal.
m	allocate nodal files according to the Modified First Fit allocation algorithm.
f	allocate nodal files according to the Fragmented Memory allocation algorithm.
l	allocate nodal files according to the Low Packing allocation algorithm.
h	allocate nodal files according to the High Packing allocation algorithm.
i	print informatory messages that detail the state of the allocation.
s	suppress creation of the l.out core image file.
u	indicate that the next option (in the form of -value) should be interpreted as the value to assign to unallocated memory in the core image file.
o	have the Allocator prompt the user for the target machine address space. The default values are those specified in the "space" declaration of the Interpreter produced file "file.a". The Allocator will attempt to allocate code only in the specified regions.
t	perform a statement trace of the allocation process. This option is useful when debugging new machine descriptions.

FILES

file.a : parsed file produced by the interpreter
file1.n : nodal output file produced by metaMicro
l.out : output file of allocator

SEE ALSO

inter(nmpc), mas(nmpc), micro(nmpc), mdump(nmpc),
Linking/Loader User's Manual, Version 1.1

NAME

ec - ecologist and smp control program

SYNOPSIS

ec [-options] [+ options] root_name optional_directory

DESCRIPTION

ec lets a user run the two simulation creation programs through one controlling program. Options beginning with a '-' are sent to the ecologist, while those beginning with a '+' go to smp. It is also possible to have ec run 'nmake', the N.mPc version of 'make'.

The ecologist is driven by a 'topology' file describing the hardware and software components of a particular simulation. The ecologist itself loads the needed ISP modules, and, if no errors occured, creates a memory list file to drive smp. smp examines the memories involved and creates memory image files in a format suitable for simulation. Because simulation preparation is decomposed into two parts, it is possible to modify a simulation by only changing the component updated, either hardware through the ecologist, or software through the smp.

When ec is run, it asks the user if the software, hardware, or both are to be changed. If only hardware is to be changed, ec runs the ecologist. If only software is to be changed, only smp is run. Changing both implies running both the ecologist and smp. If errors occured in running the ecologist, smp will not be run. If the user does not want to be prompted, the 'H', 'S', or 'B' option may be included in the '-' options going to the ecologist. 'H' implies changing only hardware, 'S' software, and 'B', both.

The simulation topology file is in a file 'root_name.t'. If a simulation is created through ec, the executable simulation will reside in a file called 'root_name', with no extensions.

Because of the file naming convention, it is most convenient to place a simulation in its own directory.

The 'option_directory' is passed to the ecologist, and is used to specify other places to find ISP compiled files. If an 'L' is encountered in the '-' options, the other directory is set to the nmpc isp' library directory.

OPTIONS

Options prefaced by a '-' go to the ecologist, and those prefaced with a '+' go to smp. To run 'nmake' over the topology file before it is given to the ecologist, add the 'M' option to the ecologist options (-M). The other two nmake options ('V' and 'U') can then be added to the invocation options for the ecologist, and they will be sent to nmake, if the 'M' option is also present.

FILES

root_name.t : topology input file
root_name.f : memory list file produced by the ecologist
root_name.x : global label and memory list file produced by the smp
root_name : the executable simulation
/nmpc/islib: ISP library directory

SEE ALSO

ecologist(nmpc), smp(nmpc), nmake(nmpc),
Ecologist User's Manual, Version 1.0

BUGS

Only one instance of '-' or '+' options may occur.

NAME

ecologist - nmpc simulation topology parser

SYNOPSIS

ecologist [-options] root_name [other_dir]

DESCRIPTION

The ecologist parses a file containing a description of the hardware and software items that make up a given simulation. The hardware modules are ISP' output files, and the software modules are core images created by the Linking/Loader. The specified hardware modules are loaded into a program containing a kernel program which creates a complete simulation program. A list of the memory images specified is placed into a file which the Simulated Memory Processor (smp) may examine to process the named memories for simulation.

The file which describes the hardware and software arrangement is called the topology file. Each simulation has a root_name. Several files exist which have the root_name, with various extensions. The topology file has the root_name with a '.t' extension. Because of the naming conventions used, it is usually desirable to place each simulation in its own directory.

If no errors were encountered in parsing the topology file, the executable program will be in a file with the root_name and no extensions.

The option 'other_dir' on the invocation line is used in front of specified ISP' output file names as a mechanism to facilitate keeping ISP' hardware modules in library directories.

OPTIONS

l	list the topology input as it is being parsed.
s	list runtime symbol file.
m	list memory files used.
t	do not remove temporary files (debugging option).
i	build a simulation which runs in separate instruction and data space (for 11/45, 55, and 70 CPU's).
f	try and link simulation with one loader call; speeds up the simulation building process. If using this option results in a loader error message, do not use it (!).
1-9	link in an alternate runtime kernel. The single character 1 to 9 will be appended to the kernel name (kernel.a), forming a new kernel name. This is useful in environments in which there are multiple kernels, each used for different purposes. Specifying a new kernel which does not exist will result in a loader error.

FILES

root_name : executable simulation program
root_name.t : topology file for simulation
root_name.f : memory list file
root_name.x : smp output file with memory names and global labels
root_name.s : symbol table
/nmpc/bin/kernel.a : runtime kernel archive

SEE ALSO

ec(nmpc), ic(nmpc), smp(nmpc),
Ecologist User's Manual

BUGS

--

NAME

ic - ISP' Compiler

SYNOPSIS

ic [-options] name.isp

DESCRIPTION

ic parses source programs in the ISP' language producing PDP-11 code to be run under the N.mPc runtime kernel.

Each source file must be terminated with a '.isp' extension. The compiler will produce an output file with the same root name, but with a '.sm' extension.

OPTIONS

l	generate a listing.
p	parse only (generate no code)
s	assembly listing of code generated in normal listing.
t	print table of ISP' structures used.
w	suppress warning messages.
T	turn on trace option.

FILES

name.isp : ISP' source file name
name.sm : ISP' output file name
/nmpc/bin/libisp.a : ISP' runtime library

SEE ALSO

ecologist(nmpc),
nmake(nmpc),
ISP' User's Manual

BUGS

Only one source program may be compiled at a time.

NAME

inter – Linking/Loader Command Program Interpreter

SYNOPSIS

inter [-options] file.i

DESCRIPTION

Inter is the name of the Linking/Loader Command Program Interpreter, a program which translates a user description of the address resolution process for a particular machine. Inter produces a file with the same root name as the input file, but with a ".a" extension. This ".a" file drives the Allocator (cater), the program which actually links metaMicro nodal output files, producing executable core image files.

OPTIONS

l	force a listing of the command program.
i	invoke the "l" option and produce a listing of included files.
s	suppress the creation of the ".a" file (syntax pass only).
p	force a listing of the contents of the command program and number statements in the mode and tranfer sections. This option is most useful when using the allocator statement trace facility (-t).

FILES

root.I : input file to be filtered through the C preprocessor, making the root.i file.
root.i : input file to be interpreted.
root.a : interpreted file to be input to the Allocator.

SEE ALSO

cater(nmpc), mas(nmpc), micro(nmpc),
Linking/Loader User's Manual, Version 1.1

BUGS

–

NAME

mas – Micro ASsembler metaMicro, Linking/Loader, Merge Interface

SYNOPSIS

mas [[-][+]options] [machine] [files.m] [files.n] [files.core]

DESCRIPTION

Through mas, a user may execute metaMicro, the Interpreter, the Allocator, and Merge, the four software generation components of N.mPc. The destination of the options depends on the intended use of mas.

To Assemble Programs
If mas is to be used to run only metaMicro, options should begin with "-" and an arbitrary number of ".m" or metaMicro source files may be specified. One of the options that must be used is the "-c" option which, as in the C compiler, instructs the software to produce only object files (".n"). Assembling 3 files to produce 3 nodal (object) files would appear as

```
mas -cXXX file1.m file2.m file3.m
```

XXX are other required metaMicro options.

To Interpret Files
Mas can be used to run the interpreter by using the following format:

```
mas [-options] machine
```

The options here are sent to the Interpreter. Machine is the name of the Interpreter input file, without the ".i" extension.

Mas will examine the directory where Linking/Loader machine descriptions are stored for a file with the name "machine.a". If this file is found, it is assumed to be the current Interpreter output and mas terminates.

If the file cannot be found, mas constructs the name "machine.I" and attempts to find that file. If this file can be found, the C preprocessor is used to filter the file into another file named "machine.i". If the "machine.I" file cannot be found, C preprocessor filtering is not performed and no errors are produced.

Mas then constructs the name "machine.i" and sends this name to the Interpreter to be interpreted.

The directory where the Linking/Loader command files are stored is installation dependent, but can be overridden with the "-I" option. For installations running Version 7 Unix, the shell variable "IPATH" can be set and exported. Mas will use "IPATH"s" value unless overridden with "-I". Also, the shell variable "MACHINE" can be set and exported and mas will use its value as the "machine" name argument noted above.

To Link Files
Mas can be called to run the Allocator and produce an Lout file. The format is

```
mas [+options] machine file1.n [file2.n ... file10.n]
```

The "+" options are sent to the Allocator. The "machine" name argument is treated as above in the section entitled "To Interpret Files".

If one argument is specified with a ".core" extension, the Lout file is renamed "file.core".

Mas can be used in combinations of the above descriptions. For example, consider a core image which is built from three metaMicro programs. Assume two are current and one has been changed. Mas can be used to run metaMicro over the modified file, and then call the Allocator to perform the linking. Format:

NAME

merge – Merge metaMicro listing files with Allocator core image files

SYNOPSIS

merge [-options] [files.core]

DESCRIPTION

merge merges listing files created by metaMicro with the information contained in the core image file produced by the Allocator, producing final listing files.

The Allocator builds the core image file with all information necessary to access all files required in final listing file production.

OPTIONS

D	display physical addresses in decimal and instruction values in octal.
X	display physical addresses and instruction values in hexadecimal.
O	display physical addresses and instruction values in octal. This is the default radix.

USAGE

Merge merges all core images files specified as arguments (or l.out if no files are specified). The merging operation involves:

1. Parsing listing files (file.l) produced by metaMicro,
2. Extracting logical addresses imbedded in those files (added to a metaMicro listing by specifying -p) and mapping to physical addressing (using the logical to physical map imbedded in the core image file),
3. Determining the length of the associated instruction (by reading the file.n nodal output file produced by metaMicro),
4. Extracting the final instruction value (from the core image file specified as an argument or l.out by default), and
5. Adding the physical address and instruction value to the listing file to produce the final listing file (file.L).

FILES

l.out : default core image file if no .core files are specified.
file.l : listing files produced by metaMicro with a -p option.
file.n : nodal files produced by metaMicro.
file.L : final listing files produced by merge.

SEE ALSO

micro(nmpc), cater(nmpc), mas(nmpc),
metaMicro-Linking/Loader Utilities User's Manual, Version 2.0

DIAGNOSTICS

Designed (hopefully) to be self-explanatory. Merge really gets confused if one of the listing files, nodal files, or core image file is out of date with the other files. The error messages produced don't make sense, start all over with metaMicro and the Allocator before merging again.

BUGS

–

NAME

micro – metaMicro Assembler

SYNOPSIS

micro [-options] file.m

DESCRIPTION

micro is the metaMicro assembler, a general microprocessor assembler which is user programmed for each assembly. metaMicro output files have the same name as the input file, except that the ".m" extension is changed to a ".n" extension. Nodal files produced by metaMicro may be link/loaded by cater, a generalized linking/loader.

OPTIONS

l	generate a source listing, with decimal line numbers.
i	invoke the "l" option and list the source code in included files.
e	envoke the "l" option and expand macros in the instruction section.
E	invoke the "l" option and expand macros in both the declaration and instruction sections.
f	invoke the "l" option and show how "if" statements cause text redirection.
p	place option. Invoke the "l" option and display the logical address of each instruction in the file. This option is required if "merge" is going to postprocess metaMicro source files.
a	invoke the "l", "e", "i", and "f" options.

USAGE

metaMicro's output should be redirected to a file named "file.l" for merge to postprocess.

FILES

file.m : metaMicro input file.
file.n : metaMicro object code output file.
file.l : metaMicro listing file, for merge. /tmp/microXXXXX : temporary file for macros.

SEE ALSO

mas(nmpc), inter(nmpc), cater(nmpc), merge(nmpc),
metaMicro User's Manual, Version 3.1
Linking/Loader User's Manual, Version 1.1
metaMicro-Linking/Loader Utilities User's Manual, Version 2.0

BUGS

Only one file may be assembled per invocation.

NAME

smp - Simulated Memory Processor

SYNOPSIS

smp [-options] memory_list.f

DESCRIPTION

smp takes core image files produced by the allocator (l.out) and prepares them for simulation by creating fixed size pages suitable for swapping at simulation time. It also collects globally defined labels into a common file to be used by the simulation and the Simulated Memory Editor (sme).

smp is driven by a memory list file produced by the ecologist. This file contains a list of the memories to be used in the given simulation. This file has the root name of the simulation, with a '.f' extension.

smp processes each file mentioned in the memory list file, leaving the output in a file with the same name as the input, except for the added '.p' extension. This '.p' file is used by the simulation.

The file containing the global symbols found in each memory list file has the simulation root name, with a '.x' extension.

smp must be run after each change in simulation software. smp may be run directly, or through the 'ec' program.

OPTIONS

l	generate a listing of smp's actions.
g	generate a listing including global labels found in each memory.
H	labels shown in the g option have addresses in hex.
D	label addresses are in decimal (default is octal)
a	page size is 32 words.
b	page size is 64 words.
c	page size is 128 words.
d	page size is 256 words (default size).
e	page size is 512 words.
f	page size is 1024 words.
s	user specied page size (format sXXX, XXX is a number from 0 to 5000. must be last option).

FILES

root_name.f : memory list file
root_name.x : processed memory list with global labels
l.out : file produced by linking loader allocator
file_name.p : file processed by smp

SEE ALSO

ec(nmpc), mas(nmpc), ecologist(nmpc),
Ecologists User's Manual Version 1.0

BUGS

–

A Simple Vax N.mPc Post Processor (V1.7)

Greg Ordy

Dept. of Computer Engr. and Science
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

This manual describes a simple post processor for the VAX implementations of the N.mPc System. It will run under both the VMS and Unix (4.1 BSD) operating systems. This post processor is considerably simpler than the one written for the PDP-11 implementations, however it performs the basic function of reading the data files (*.d) produced by the simulation, and displaying its contents. The post processor is written in the C language, as is most of N.mPc.

1. Introduction

The VAX post processor is named 'pp', as is the PDP-11 version. When running a simulation, the trace command can be used to generate time-tagged data which is placed in a file with the root simulation name, and a '.d' suffix. Each unique trace command is tagged with a monitor number which is the number returned when the trace command is executed. For example, here is a fragment from an example simulation:

```
# ...
# ...
# trace cpu:ir
monitor number 3
# ...
# ...
```

In response to the trace command request the command interpreter named that monitor '#3'. That number (3) is also used to refer to that traced data stream in the post processor.

When running the post processor, any set, or all of the data traces may be displayed. In addition, the output base may be set to any one of either 2, 8, 10, or 16. Note that in all cases simulation time is always displayed in base 10.

Because this program was quickly designed to be a temporary replacement for a more powerful post processor, there are some limitations.

1) Any structure which is wider than 32 bits cannot be displayed as a decimal number.

2) Simulation time is only maintained to a precision of 32 bits.

Even with these restrictions, 'pp' should be quite useful.

2. Post Processor Options

The name of the post processor is 'pp'. Each invocation must have the name of the trace data file to be examined specified. Note that 'pp' will accept names with or without the '.d' suffix. In addition to the data file name, one or more program options may be specified.

In addition to the basic function of displaying trace data, the post processor summerizes the data found in the file in terms of the number of entries of each monitor. When the post processor is finished displaying data, a summary will be displayed.

2.1. Number Base Selection

The default output base for all data values is base 16. Options to modify this are:

-b	Display data in base 2 (binary).
-o	Display data in base 8 (octal).
-d	Display data in base 10 (decimal).
-x	Display data in base 16 (hex).

For bases other than 2, all leading zeros are removed from data values. Base 2 data is shown at the full field size, rounded up to the next multiple of 8 bits.

2.2. Monitor Specification

By default, all monitors encountered in the trace file are displayed. The set may be reduced by explicitly specifying the monitors to be shown. Each monitor number is specified as '#x', where 'x' is the monitor number. If any monitor numbers are specified, all unspecified monitors are automatically disregarded.

2.3. Other Options

The post processor also supports the following options:

-s	Only generate a summary, no data display.
-n	Do not generate a summary.
-f{name}	Generate output into a file named 'name'. This option is most useful on VMS systems. Note that error information will still be sent to the terminal. The file name is directly concatenated to the 'f' option character. For example, if you want the output to go into a file named 'data.txt', you would specify to 'pp':

```
pp -fdata.txt ... ... ...
```

Other options may not be used in the same option string as the '-f' option, after the file name.

3. Misc.

If no options are specified, the default is to display all monitors in base 16, as well as the data summary. Displayed time values, as well as data in the summary section is always displayed in decimal.

Example:

```
pp -d siml
```

Display all monitors in the data file 'siml.d' in base 10. A summary will be printed at the end.

Example:

```
pp -b #3 #12 try.d
```

Display data for only monitors 3 and 12 in file 'try.d' in base 2. A summary will be printed at the end.

Simulation time is displayed whenever it changes relative to the display of monitors, and to determine the time that a displayed value occurred, just look backwards to find the last time entry.

Due to the implementation of the runtime command interpreter, names of display commands will also be placed in the trace data file. Their data, however, is shown on the terminal, and is not placed in the file.

Currently, all monitor data is shown in the same number base. The internal structure of the program is set up to allow a separate base per monitor. To implement per monitor display bases all that need be changed is the portion of the program which deals with parsing program options.

Sample Post Processor Output

```
[illegible]
created Fri Jul 13 12:56:33 1984
output to: simdata
-------------------------------
All monitors displayed.
Monitors displayed in base 2.
-------------------------------
Establish #2: 'trace :D[1] read' at t = 0
Establish #3: 'trace :D[2] read' at t = 0
Establish #4: 'trace :A[0] read' at t = 0
Establish #5: 'trace :PC read' at t = 0
Establish #6: 'trace :SR read' at t = 0
Establish #7: 'trace :A1 read' at t = 0
Establish #8: 'trace :A2 read' at t = 0
Establish #9: 'trace :D1 read' at t = 0
Establish #10: 'trace :D2 read' at t = 0
Establish #11: 'trace :I1 read' at t = 0
Establish #12: 'trace :I2 read' at t = 0
Establish #13: 'trace :S1 read' at t = 0
Establish #14: 'trace :ADDRESS' at t = 0
Establish #15: 'trace mem:ADDRESS read' at t = 0
Establish #16: 'trace :AS' at t = 0
Establish #17: 'trace mem:AS read' at t = 0
Establish #18: 'trace :UDS' at t = 0
Establish #19: 'trace mem:UDS read' at t = 0
Establish #20: 'trace :LDS' at t = 0
Establish #21: 'trace mem:LDS read' at t = 0
Establish #22: 'trace :DTACK read' at t = 0
Establish #23: 'trace mem:DTACK' at t = 0
Establish #24: 'trace :R_W' at t = 0
Establish #25: 'trace mem:R_W read' at t = 0
Establish #26: 'trace :FC' at t = 0
Establish #27: 'trace mem:FC read' at t = 0
Establish #28: 'trace :DATA read' at t = 0
Establish #29: 'trace mem:DATA read' at t = 0
            #24 'trace :R_W' = 00000001
            #20 'trace :LDS' = 00000001
            #18 'trace :UDS' = 00000001
            #16 'trace :AS' = 00000001
            #24 'trace :R_W' = 00000001
t = 60
            #26 'trace :FC' = 00000010
            #14 'trace :ADDRESS' = 000000000000100000000000
t = 120
            #20 'trace :LDS' = 00000000
            #18 'trace :UDS' = 00000000
            #16 'trace :AS' = 00000000
            #29 'trace mem:DATA read' = 0000000000000001
            #29 'trace mem:DATA read' = 0011010000000001
            #23 'trace mem:DTACK' = 00000001
t = 420
            #20 'trace :LDS' = 00000001
            #18 'trace :UDS' = 00000001
            #16 'trace :AS' = 00000001
            #29 'trace mem:DATA read' = 0000000000000000
            #23 'trace mem:DTACK' = 00000000
t = 480
            #26 'trace :FC' = 00000000
            #14 'trace :ADDRESS' = 000000000000000000000000
            #11 'trace :I1 read' = 0011010000000001
            #5 'trace :PC read' = 00000000000000000001000000000010
            #9 'trace :D1 read' = 0000000000000000000000000101010101010101
            #6 'trace :SR read' = 0000000000000000
            #6 'trace :SR read' = 0000000000000100
            #3 'trace :D[2] read' = 000000000000000000101010101010101
            #6 'trace :SR read' = 0000000000000100
            #6 'trace :SR read' = 0000000000000100
t = 540
            #24 'trace :R_W' = 00000001
t = 600
            #26 'trace :FC' = 00000010
            #14 'trace :ADDRESS' = 000000000000100000000001
```

```
t = 660
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 960
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 1020
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000000001000000000100
        #9 'trace :D1 read' = 0000000000000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
t = 1080
        #24 'trace :R_W' = 00000001
t = 1140
        #26 'trace :FC' = 00000010
        #14 'trace :ADDRESS' = 000000000000100000000010
t = 1200
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 1500
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 1560
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000000001000000000110
        #9 'trace :D1 read' = 0000000000000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
t = 1620

        #24 'trace :R_W' = 00000001
t = 1680
        #26 'trace :FC' = 00000010
        #14 'trace :ADDRESS' = 000000000000100000000011
t = 1740
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 2040
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 2100
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000000001000000001000
        #9 'trace :D1 read' = 0000000000000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
```

```
t = 216Ø
         #24 'trace :R_W' = ØØØØØØØ1
t = 222Ø
         #26 'trace :FC' = ØØØØØØ1Ø
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØ1ØØØØØØØØ1ØØ
t = 228Ø
         #2Ø 'trace :LDS' = ØØØØØØØØ
         #18 'trace :UDS' = ØØØØCØØØ
         #16 'trace :AS' = ØØØØØØØØ
         #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØ1
         #29 'trace mem:DATA read' = ØØ11Ø1ØØØØØØØØØ1
         #23 'trace mem:DTACK' = ØØØØØØØ1
t = 25ØØ
         #2Ø 'trace :LDS' = ØØØØØØØ1
         #18 'trace :UDS' = ØØØØØØØ1
         #16 'trace :AS' = ØØØØØØØ1
         #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØØ
         #23 'trace mem:DTACK' = ØØØØØØØØ
t = 264Ø
         #26 'trace :FC' = ØØØØØØØØ
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØØØØØØØØØØØØØ
         #11 'trace :I1 read' = ØØ11Ø1ØØØØØØØØØ1
         #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØ1Ø1Ø
         #9 'trace :D1 read' = ØØØØØØØØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #3 'trace :D[2] read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
t = 27ØØ
         #24 'trace :R_W' = ØØØØØØØ1
t = 276Ø
         #26 'trace :FC' = ØØØØØØ1Ø
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØ1ØØØØØØØØ1Ø1
t = 282Ø
         #2Ø 'trace :LDS' = ØØØØØØØØ

         #18 'trace :UDS' = ØØØØØØØØ
         #16 'trace :AS' = ØØØØØØØØ
         #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØ1
         #29 'trace mem:DATA read' = ØØ11Ø1ØØØØØØØØØ1
         #23 'trace mem:DTACK' = ØØØØØØØ1
t = 312Ø
         #2Ø 'trace :LDS' = ØØØØØØØ1
         #18 'trace :UDS' = ØØØØØØØ1
         #16 'trace :AS' = ØØØØØØØ1
         #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØØ
         #23 'trace mem:DTACK' = ØØØØØØØØ
t = 318Ø
         #26 'trace :FC' = ØØØØØØØØ
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØØØØØØØØØØØØØ
         #11 'trace :I1 read' = ØØ11Ø1ØØØØØØØØØ1
         #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØ11ØØ
         #9 'trace :D1 read' = ØØØØØØØØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #3 'trace :D[2] read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
         #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
t = 324Ø
         #24 'trace :R_W' = ØØØØØØØ1
t = 33ØØ
         #26 'trace :FC' = ØØØØØØ1Ø
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØ1ØØØØØØØØ11Ø
t = 336Ø
         #2Ø 'trace :LDS' = ØØØØØØØØ
         #1Ø 'trace :UDS' = ØØØØØØØØ
         #16 'trace :AS' = ØØØØØØØØ
         #29 'trace mem:DATA read' = ØØØØØØØØ11Ø1ØØØØ
         #29 'trace mem:DATA read' = Ø1ØØ111Ø11Ø1ØØØØ
         #23 'trace mem:DTACK' = ØØØØØØØ1
t = 366Ø
         #2Ø 'trace :LDS' = ØØØØØØØ1
         #18 'trace :UDS' = ØØØØØØØ1
         #16 'trace :AS' = ØØØØØØØ1
         #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØØ
         #23 'trace mem:DTACK' = ØØØØØØØØ
t = 372Ø
         #26 'trace :FC' = ØØØØØØØØ
         #14 'trace :ADDRESS' = ØØØØØØØØØØØØØØØØØØØØØØØØ
         #11 'trace :I1 read' = Ø1ØØ111Ø11Ø1ØØØØ
         #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØ111Ø
         #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØØØØØ
```

```
t = 3780
        #24 'trace :R_W' = 00000001
t = 3840
        #26 'trace :FC' = 00000010
        #14 'trace :ADDRESS' = 000000000000100000000000
t = 3960
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 4200
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 4260
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000001000000000000010
        #9 'trace :D1 read' = 00000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
t = 4320
        #24 'trace :R_W' = 00000001
t = 4380
        #26 'trace :FC' = 00000010
        #14 'trace :ADDRESS' = 000000000000100000000001
t = 4440
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 4740
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 4800
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000001000000000000100
        #9 'trace :D1 read' = 00000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
t = 4860
        #24 'trace :R_W' = 00000001
t = 4920
        #26 'trace :FC' = 00000010
        #14 'trace :ADDRESS' = 000000000000100000000010
t = 4980
        #20 'trace :LDS' = 00000000
        #18 'trace :UDS' = 00000000
        #16 'trace :AS' = 00000000
        #29 'trace mem:DATA read' = 0000000000000001
        #29 'trace mem:DATA read' = 0011010000000001
        #23 'trace mem:DTACK' = 00000001
t = 5280
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 5340
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :I1 read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000001000000000000110
        #9 'trace :D1 read' = 00000000000000000101010101010101
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #3 'trace :D[2] read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
t = 54ØØ
        #24 'trace :R_W' = ØØØØØØØ1
t = 546Ø
        #26 'trace :FC' = ØØØØØØ1Ø
        #14 'trace :ADDRESS' = ØØØØØØØØØØØ1ØØØØØØØØØ11
t = 552Ø
        #2Ø 'trace :LDS' = ØØØØØØØØ
        #18 'trace :UDS' = ØØØØØØØØ
        #16 'trace :AS' = ØØØØØØØØ
        #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØ1
        #29 'trace mem:DATA read' = ØØ11Ø1ØØØØØØØØØ1
        #23 'trace mem:DTACK' = ØØØØØØØ1
t = 582Ø
        #2Ø 'trace :LDS' = ØØØØØØØ1
        #18 'trace :UDS' = ØØØØØØØ1
        #16 'trace :AS' = ØØØØØØØ1
        #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØØ
        #23 'trace mem:DTACK' = ØØØØØØØØ
t = 588Ø
        #26 'trace :FC' = ØØØØØØØØ
        #14 'trace :ADDRESS' = ØØØØØØØØØØØØØØØØØØØØØØØ
        #11 'trace :I1 read' = ØØ11Ø1ØØØØØØØØØ1
        #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØ1ØØØ
        #9 'trace :D1 read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #3 'trace :D[2] read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
t = 594Ø
        #24 'trace :R_W' = ØØØØØØØ1
t = 6ØØØ
        #26 'trace :FC' = ØØØØØØ1Ø
        #14 'trace :ADDRESS' = ØØØØØØØØØØØ1ØØØØØØØØ1ØØ
t = 6Ø6Ø
        #2Ø 'trace :LDS' = ØØØØØØØØ
        #18 'trace :UDS' = ØØØØØØØØ
        #16 'trace :AS' = ØØØØØØØØ
        #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØ1
        #29 'trace mem:DATA read' = ØØ11Ø1ØØØØØØØØØ1
        #23 'trace mem:DTACK' = ØØØØØØØ1
t = 636Ø
        #2Ø 'trace :LDS' = ØØØØØØØ1
        #18 'trace :UDS' = ØØØØØØØ1
        #16 'trace :AS' = ØØØØØØØ1
        #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØØ
        #23 'trace mem:DTACK' = ØØØØØØØØ
t = 642Ø
        #26 'trace :FC' = ØØØØØØØØ
        #14 'trace :ADDRESS' = ØØØØØØØØØØØØØØØØØØØØØØØ
        #11 'trace :I1 read' = ØØ11Ø1ØØØØØØØØØ1
        #5 'trace :PC read' = ØØØØØØØØØØØØØØØØØØØ1ØØØØØØØØ1Ø1Ø
        #9 'trace :D1 read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #3 'trace :D[2] read' = ØØØØØØØØØØØØØØØØØ1Ø1Ø1Ø1Ø1Ø1Ø1Ø1
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
        #6 'trace :SR read' = ØØØØØØØØØØØØØ1ØØ
t = 646Ø


        #24 'trace :R_W' = ØØØØØØØ1
t = 654Ø
        #26 'trace :FC' = ØØØØØØ1Ø
        #14 'trace :ADDRESS' = ØØØØØØØØØØØ1ØØØØØØØØ1Ø1
t = 66ØØ
        #2Ø 'trace :LDS' = ØØØØØØØØ
        #18 'trace :UDS' = ØØØØØØØØ
        #16 'trace :AS' = ØØØØØØØØ
        #29 'trace mem:DATA read' = ØØØØØØØØØØØØØØØ1
        #29 'trace mem:DATA read' = ØØ11Ø1ØØØØØØØØØ1
        #23 'trace mem:DTACK' = ØØØØØØØ1
```

```
t = 6900
        #20 'trace :LDS' = 00000001
        #18 'trace :UDS' = 00000001
        #16 'trace :AS' = 00000001
        #29 'trace mem:DATA read' = 0000000000000000
        #23 'trace mem:DTACK' = 00000000
t = 6960
        #26 'trace :FC' = 00000000
        #14 'trace :ADDRESS' = 000000000000000000000000
        #11 'trace :II read' = 0011010000000001
        #5 'trace :PC read' = 00000000000000000000010000001100
        #9 'trace :D1 read' = 00000000000000000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
        #3 'trace :D[2] read' = 00000000000000000000101010101010101
        #6 'trace :SR read' = 0000000000000100
        #6 'trace :SR read' = 0000000000000100
--------------------------------

Summary of trace data:
#3: 'trace :D[2] read' 12 entries.
#5: 'trace :PC read' 14 entries.
#6: 'trace :SR read' 48 entries.
#9: 'trace :D1 read' 12 entries.
#11: 'trace :II read' 13 entries.
#14: 'trace :ADDRESS' 26 entries.
#16: 'trace :AS' 27 entries.
#18: 'trace :UDS' 27 entries.
#20: 'trace :LDS' 27 entries.
#23: 'trace mem:DTACK' 26 entries.
#24: 'trace :R_W' 14 entries.
#26: 'trace :FC' 26 entries.
#29: 'trace mem:DATA read' 39 entries.
```

Appendix B: ISP'/CDL Declaration Sections

```
/**********************************************************************/
/*                                                                    */
/*              MC68000 ISP' Model Structure Declarations             */
/*                                                                    */
/**********************************************************************/

state

/**********************************************************************/
/*                                                                    */
/*                 M68000 Programming Registers                       */
/*                                                                    */
/**********************************************************************/

D[0:7]<31:0>,                  ! 8 Data Registers
A[0:6]<31:0>,                  ! 7 Address Registers
UA7<31:0>,                     ! User Stack Pointer
SA7<31:0>,                     ! System Stack Pointer
PC<31:0>,                      ! Program Counter
SR<15:0>,                      ! Status Register

/**********************************************************************/
/*                                                                    */
/*                 Temporary Internal Registers                       */
/*                                                                    */
/**********************************************************************/

PFR<15:0>,                     ! Prefetch Register
IR<15:0>,                      ! Instruction Register
FC<2:0>,                       ! Function Code Register
EXDBUF<15:0>,                  ! External Data Bus Buffer Register
EXABUF<23:1>,                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,                 ! ALU Buffer 1
ALUBUF2<31:0>,                 ! ALU Buffer 2
DTEMP<15:0>,                   ! Temporary Data Storage
DISREG<31:0>,                  ! Temporary Displacement Storage
SRTEMP<15:0>,                  ! Temporary Status Register Storage
                               ! (Exception Processing)
IRTEMP<15:0>,                  ! Temporary Instruction Register Storage
                               ! (Exception Processing)
TEMPADR<31:0>,                 ! Temporary Cycle Address Storage
                               ! (Exception Processing)
ACTYPE<15:0>,                  ! Temporary Access Type Storage
                               ! (Exception Processing)
VECADR<23:0>,                  ! Temporary Vector Address Storage
                               ! (Exception Processing)
HANADR<31:0>,                  ! Temporary Address Storage For
                               ! Exception Handler Routine
```

```
T<7:0>,                         ! Clock Cycle Counter
RESET,                       ! Reset Flip-Flop
HALT,                        ! Halt Flip-Flop
RW,                          ! Read/Write Flip-Flop
ADENABLE,                    ! Address Bus Buffer Enable
DBENABLE,                    ! Data Bus Buffer Enable
ASN,                         ! Address Strobe Flip-Flop
LDSN,                        ! Lower Data Strobe Flip-Flop
UDSN,                        ! Upper Data Strobe Flip-Flop
DTACKN,                      ! Data Transfer Acknowledge Flip-Flop
COUT,                        ! Carry Flip-Flop
EXCEPT,                      ! Exception Processing Flip-Flop
READY,                       ! Ready Flip-Flop

/***********************************************************************/
/*                                                                     */
/*      Model transformation modifications:                            */
/*                                                                     */
/*           1) CDL decoder structure nonexistent in ISP' and un-      */
/*      necessary for model. Eliminated.                               */
/*           2) Multi-phase clock structure nonexistent in ISP'.       */
/*      Operations on registers will provide its equivalent.           */
/*           3) Switch structure nonexistent in ISP'. Operation on a   */
/*      register will provide its equivalent.                          */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model   */
/*      equivalency and simplicity.                                    */
/*           5) The memory word length was reduced from 16 to 8 bit    */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing     */
/*      MC68000 assembler and linker/loader models. The memory was     */
/*      also reduced from 8 Mwords to 32 Kbytes.                       */
/*                                                                     */
/***********************************************************************/

IABUS<31:0>,                    ! Internal Address Bus
IDBUS<31:0>,                    ! Internal Data Bus
SWITCH,                      ! Power Switch
PHI1,                        ! Phase 1 Of Two-Phase Clock
PHI2;                        ! Phase 2 Of Two-Phase Clock

port

/************************************************************************/
/*                                                                      */
/*             External Address and Data Bus                           */
/*                                                                      */
/************************************************************************/

DBUS<15:0>,                     ! Data Bus
ABUS<23:1>;                     ! Address Bus

format
```

```
/*************************************************************************/
/*                                                                       */
/*                        Register Subfields                             */
/*                                                                       */
/*************************************************************************/

PCADDR        = PC<23:0>,        ! Program Counter Address Field
SRTRACE       = SR<15>,          ! Trace Bit
SRMODE        = SR<13>,          ! Mode Selection Bit
SRCARRY       = SR<0>,           ! Carry Bit
SROVER        = SR<1>,           ! Overflow Bit
SRZERO        = SR<2>,           ! Zero Bit
SRNEG         = SR<3>,           ! Negative Bit
SREX          = SR<4>,           ! Extend Bit
SRMASK        = SR<10:8>,        ! Interrupt Mask
FCSPACE       = FC<1:0>,         ! Memory Access Address Space
FCMODE        = FC<2>,           ! User/Supervisor Mode Bit
PCLOW         = PC<15:0>,        ! PC Low Word
PCHI          = PC<31:16>,       ! PC High Word
D0LWORD       = D[0]<15:0>,      ! D[0] Low Word
D1LWORD       = D[1]<15:0>,      ! D[1] Low Word
D2LWORD       = D[2]<15:0>,      ! D[2] Low Word
D3LWORD       = D[3]<15:0>,      ! D[3] Low Word
D4LWORD       = D[4]<15:0>,      ! D[4] Low Word
D5LWORD       = D[5]<15:0>,      ! D[5] Low Word
D6LWORD       = D[6]<15:0>,      ! D[6] Low Word
D7LWORD       = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW     = HANADR<15:0>, ! HANADR Low Word
HANADRHI      = HANADR<31:16>,! HANADR High Word
TEMPADRLOW    = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI     = TEMPADR<31:16>;! TEMPADR High Word

memory

/*************************************************************************/
/*                                                                       */
/*                  16K 16-Bit Word Internal Memory                      */
/*                                                                       */
/*************************************************************************/

M[0:32767]<7:0>;
```

[illegible] Declarations

```
$ Programming registers
Register,     D0(0-31),             $data registers
              D1(0-31),
              D2(0-31),
              D3(0-31),
              D4(0-31),
              D5(0-31),
              D6(0-31),
              D7(0-31),
              A0(0-31),             $address registers
              A1(0-31),
              A2(0-31),
              A3(0-31),
              A4(0-31),
              A5(0-31),
              A6(0-31),
              UA7(0-31),            $user stack pointer
              SA7(0-31),            $supervisor stack
                                    pointer
              PC(0-31),             $program counter
              SR(0-15),             $status register

$ Internal registers (defined by authors)
Register,     PFR(0-15),            $prefetch register
              IR(0-15),             $instruction register
              FC(0-2),              $function code register
              EXDBUF(0-15),         $external data bus
                                    buffer register
              EXABUF(0-15),         $external address bus
                                    buffer register
              ALUBUF1(0-31),        $ALU buffer 1
              ALUBUF2(0-31),        $ALU buffer 2
              DTEMP(0-15),          $temporary storage for
                                    data
              DISREG(0-31),         $temporary storage for
                                    displacement
              SRTEMP(0-15),         $temporary storage for
                                    status register (used
                                    for exception
                                    processing)
              IRTEMP(0-15),         $temporary storage for
                                    instruction register
                                    (used for exception
                                    processing)
              TEMPADR(0-31),        $temporary storage for
                                    current cycle address
                                    (used for exception
                                    processing)
              ACTYPE(0-15),        $temporary storage for
                                   access type information
                                   (used for exception
                                   processing)
              VECADR(0-23),        $temporary storage for
                                   vector address (used for
                                   exception processing)
              HANADR(0-31),        $temporary storage for
                                   address of exception
                                   handler routine (used
                                   for exception
                                   processing)
              T(0-7),              $control register, clock
                                   cycle counter (reset at
                                   end of each instruction)
              RESET,               $reset flip-flop
              HALT,                $halt flip-flop
              RW,                  $read/write flip-flop
              ADENABLE,            $address bus buffer
                                   enable, high impedance
                                   when low, enabled when
                                   high
              DBENABLE,            $data bus buffer enable,
                                   high impedance when low,
                                   enabled when high
              ASN,                 $address strobe
                                   flip-flop
              LDSN,                $lower data strobe

                                   flip-flop
              UDSN,                $upper data strobe
                                   flip-flop
              DTACKN,              $data transfer
                                   acknowledge flip-flop
                                   (from peripheral device)
              COUT,                $carry flip-flop
              EXCEPT,              $exception processing
                                   flip-flop
              READY                $READY flip-flop
                                   (indicates processor is
                                   ready after power up or
                                   reset)

$ Subregister declarations
Subregisters, PC(ADDR)=PC(0-23),   $address field of
                                   program counter
              SR(TRACE)=SR(15),    $trace bit
              SR(MODE)=SR(13),     $mode selection bit
              SR(CARRY)=SR(0),     $carry bit
              SR(OVER)=SR(1),      $overflow bit
              SR(ZERO)=SR(2),      $zero bit
              SR(NEG)=SR(3),       $negative bit
              SR(EX)=SR(4),        $extend bit
```

```
                     SR(MASK)=SR(8-10),    $interrupt mask
                     FC(SPACE)=FC(0-1),    $address space of memory
                                           access
                     FC(MODE)=FC(2),       $user or supervisor mode
                                           bit
                     PC(LOW)=PC(0-15),     $low word of PC
                     PC(HI)=PC(16-31),     $high word of PC
                     D0(LWORD)=D0(0-15),   $low word of D0
                     D1(LWORD)=D1(0-15),   $low word of D1
                     D2(LWORD)=D2(0-15),   $low word of D2
                     D3(LWORD)=D3(0-15),   $low word of D3
                     D4(LWORD)=D4(0-15),   $low word of D4
                     D5(LWORD)=D5(0-15),   $low word of D5
                     D6(LWORD)=D6(0-15),   $low word of D6
                     D7(LWORD)=D7(0-15),   $low word of D7
                     DISREG(HWORD)=        $high word of DISREG
                         DISREG(16-31)
                     DISREG(LWORD)=        $low word of DISREG
                     HANADR(LOW)=          $low word of HANADR
                         HANADR(0-15),
                     HANADR(HI)=           $high word of HANADR
                         HANADR(16-31),
                     TEMPADR(LOW)=         $low word of TEMPADR
                         TEMPADR(0-15),
                     TEMPADR(HI)=          $high word of TEMPADR
                         TEMPADR(16-31)

$ Bus declarations
Bus,                 IABUS(0-31),          $internal address bus
                     IDBUS(0-31),          $internal data bus
                     DBUS(0-15),           $external data bus
                     ABUS(0-23)            $external address bus

$ Decoder declarations
Decoders,            A(0-3)=IR(14-15),     $decoders used to decode
                                           various fields of the
                                           instruction register
                     B(0-3)=IR(12-13),
                     C(0-7)=IR(9-11),
                     D(0-7)=IR(6-8),
                     E(0-7)=IR(3-5),
                     F(0-7)=IR(0-2),
                     G(0-15)=IR(8-11),
                     H(0-3)=IR(6-7),
                     K(0-256)=T(0-7)       $control register (clock
                                           cycle counter) decoder

$ Switch, memory, and clock declarations
Switch,              POWER(ON,OFF)         $power switch

Memory,              M( )=M(0-8388608,0-15)

Clock,               P(1-2)                $two phase clock
```

Appendix C: ISP' Models of MC68000 Instructions

The ISP' descriptions for each of the modeled MC68000 instructions or exception sequences appear in this appendix. They are:

41

21. ILLEGAL ADDRESS EXCEPTION C-358

```
/*********************************************************************/
/*                                                                   */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W D1,D2 INSTRUCTION        */
/*                                                                   */
/*********************************************************************/

/*********************************************************************/
/*                                                                   */
/*                      Structure Declarations                       */
/*                                                                   */
/*********************************************************************/

state

/*********************************************************************/
/*                                                                   */
/*                  M68000 Programming Registers                     */
/*                                                                   */
/*********************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/*********************************************************************/
/*                                                                   */
/*                  Temporary Internal Registers                     */
/*                                                                   */
/*********************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,               ! Temporary Address Storage For
                            ! Exception Handler Routine
T<7:0>,                     ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*    Model transformation modifications:                            */
/*                                                                    */
/*          1) CDL decoder structure nonexistent in ISP' and un-      */
/*    necessary for model. Eliminated.                                */
/*          2) Multi-phase clock structure nonexistent in ISP'.       */
/*    Operations on registers will provide its equivalent.            */
/*          3) Switch structure nonexistent in ISP'. Operation on a   */
/*    register will provide its equivalent.                           */
/*          4) The declared bus structures are modeled with registers */
/*    without loss of model accurracy. This done to maintain model    */
/*    equivalency and simplicity.                                     */
/*          5) The memory word length was reduced from 16 to 8 bit    */
/*    words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*    their PC incrementation, and to enable the use of existing      */
/*    MC68000 assembler and linker/loader models. The memory was      */
/*    also reduced from 8 Mwords to 32 Kbytes.                        */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,                ! Internal Address Bus
IDBUS<31:0>,                ! Internal Data Bus
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                 ! External Data Bus
ABUS<23:1>;                 ! External Address Bus(changed)
```

```
format

/******************************************************************************/
/*                                                                            */
/*                        Register Subfields                                  */
/*                                                                            */
/******************************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/******************************************************************************/
/*                                                                            */
/*                    16K 16-Bit Word Internal Memory                         */
/*                                                                            */
/******************************************************************************/

M[0:32767]<7:0>;

macro

/******************************************************************************/
/*                                                                            */
/*                        Logic Level Macros                                  */
/*                                                                            */
```

```
/*************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[4110] = 0xff;                 ! Place Memory Locations Following The
        M[4111] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*     Routine Initialization Per Hamby and Guillory             */
        /*                                                               */
        /*****************************************************************/
        SRMODE = lo;                    ! Set Status Register To User Mode
        D[1] = 0x55555555;              ! Place Hex 55555555 Into D[1]
        A[0] = 0x1000;                  ! Place Hex 1000 Into A[0]
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/*************************************************************************/
/*                                                                       */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary     */
/*  to retrieve modeled instructions for simulation and analysis. It     */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                    */
/*                                                                     */
/*********************************************************************/

fetch_initial_instruction :=
       (

       /*****************************************************************/

       PHI1 = hi;                               ! Phase 1 Of
       PHI2 = lo;                               ! Clock Cycle 0
       RW = hi;                                 ! Memory Read
       ADENABLE = lo;                           ! Disable Address Bus Buffer
       DBENABLE = lo;                           ! Disable Data Bus Buffer
       IABUS = PC;                              ! Place PC On Internal Address
                                                ! Bus
       next;                                    ! Execute Pending Assignments

       PHI1 = lo;                               ! Phase 2 Of
       PHI2 = hi;                               ! Clock Cycle 0
       ADENABLE = hi;                           ! Enable Address Bus Buffer
       EXABUF = IABUS;                          ! Gate Internal Address Bus
                                                ! Into External Address Buffer
       FCMODE = SRMODE;                         ! User Mode
       FCSPACE = 2;                             ! Accessing Program
       next;                                    ! Execute Impending Assignments
       ABUS = EXABUF;                           ! Address Placed On Bus(Added)
       next;                                    ! Execute Pending Assignments

       /*****************************************************************/
       T = 1;                                   ! Clock Cycle 1
       next;                                    ! Execute Assignment

       PHI1 = hi;                               ! Phase 1 Of
       PHI2 = lo;                               ! Clock Cycle 1
       ASN = lo;                                ! Assert Address Strobe
       LDSN = lo;                               ! Assert Lower Data Strobe
       UDSN = lo;                               ! Assert Upper Data Strobe
       DBENABLE = hi;                           ! Enable Data Bus
       next;                                    ! Execute Pending Assignments

       PHI1 = lo;                               ! Phase 2
       PHI2 = hi;                               ! Of Clock Cycle 1
       next;                                    ! Execute Pending Assignments

       /*****************************************************************/
       T = 2;                                   ! Clock Cycle 2
       next;                                    ! Execute Assignment

       PHI1 = hi;                               ! Phase 1
       PHI2 = lo;                               ! Of Clock Cycle 2
       while DTACKN eql hi                      ! Wait For Memory To Place
             (                                  ! Data On The Bus
        next;                              ! Execute Impending Assignments

        PHI1 = lo;                         ! Phase 2
        PHI2 = hi;                         ! Of Clock Cycle 2
        next;                              ! Execute Assignments

        /**********************************************************/
        T = 3;                             ! Clock Cycle 3
        next;                              ! Execute Assignment

        PHI1 = hi;                         ! Phase 1
        PHI2 = lo;                         ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
        DTACKN = lo;                       ! Asserts DTACKN(Added)
        next;                              ! Execute Pending Assignments

        /**********************************************************/
        T = 2                              ! Return To  Phase 2
                                           ! Of Clock Cycle 2
        );
        next;                              ! Execute Impending Assignments

/*******************************************************************/
T = 3;                                     ! Clock Cycle 3
next;                                      ! Execute Assignment

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 3
EXDBUF = DBUS;                             ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
next;                                      ! Execute Pending Assignments

/*******************************************************************/
T = 4;                                     ! Clock Cycle 4
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 4
PFR = EXDBUF;                              ! The Contents Of The External
                                           ! Data Bus Buffer Are Placed
                                           ! In Prefetch Register
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 4
ASN = hi;                                  ! Deactivate Address Strobe
LDSN = hi;                                 ! Deactivate Lower Data Strobe
UDSN = hi;                                 ! Deactivate Upper Data Strobe
IR = PFR;                                  ! Contents Of Prefetch Register
                                           ! Are Placed Into Instruction
                                           ! Register
        DTACKN = hi;                          ! Deactivate Data Transfer(Added)
                                              ! Acknowledge
        PC = PC + 2;                          ! Increment Program Counter
        next;                                 ! Execute Pending Assignments
        T = 0                                 ! Reset Clock Cycle Counter
        )



move :=                                       ! MOVE.W D1,D2
        (

        /*****************************************************************/

        PHI1 = hi;                            ! Phase 1 Of
        PHI2 = lo;                            ! Clock Cycle 0
        DBUS = 0xffff;                        ! Place Data Bus In High Impedance
        RW = hi;                              ! Memory Read
        ADENABLE = lo;                        ! Disable Address Bus Buffer
        DBENABLE = lo;                        ! Disable Data Bus Buffer
        IABUS = PC;                           ! Place PC On Internal Address
                                              ! Bus
        IDBUS = D1LWORD;                      ! Place Low Word From D[1] Onto
                                              ! Internal Data Bus
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2 Of
        PHI2 = hi;                            ! Clock Cycle 0
        ADENABLE = hi;                        ! Enable Address Bus Buffer
        EXABUF = IABUS;                       ! Gate Internal Address Bus
                                              ! Into External Address Buffer
        FCMODE = SRMODE;                      ! User Mode
        FCSPACE = 2;                          ! Accessing Program
        SRCARRY = lo;                         ! Clear Status Register Carry Bit
        SROVER = lo;                          ! Clear Status Register Overflow Bit
        SRZERO = lo;                          ! Clear Status Register Zero Bit
        SRNEG  = lo;                          ! Clear Status Register Negative Bit
        D2LWORD = IDBUS;                      ! Place Data From Internal Data Bus
                                              ! Into Low Word Of D[2]
        next;                                 ! Execute Impending Assignments
        ABUS = EXABUF;                        ! Address Placed On Bus(Added)
        next;                                 ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                ! Clock Cycle 1
        next;                                 ! Execute Assignment

        PHI1 = hi;                            ! Phase 1 Of
        PHI2 = lo;                            ! Clock Cycle 1
        ASN = lo;                             ! Assert Address Strobe
        LDSN = lo;                            ! Assert Lower Data Strobe
        UDSN = lo;                            ! Assert Upper Data Strobe
        DBENABLE = hi;                        ! Enable Data Bus
```

```
if D2LWORD eql 0                        ! Set Status Register Zero Bit
   SRZERO = hi;                         ! If Moved Data Is Zero
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
if D[2]<15>                             ! Set Status Register Negative
   SRNEG = hi;                          ! Bit If Moved Data Is Negative
next;                                   ! Execute Pending Assignments

/*******************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*************************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*************************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*******************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*******************************************************************/
```

```
        T = 4;                                  ! Clock Cycle 4
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 4
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )


   jmp :=                                       ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments
```

```
/***********************************************************************/
T = 1;                                              ! Clock Cycle 1
next;                                               ! Execute Assignment

PHI1 = hi;                                          ! Phase 1 Of
PHI2 = lo;                                          ! Clock Cycle 1
ASN = lo;                                           ! Assert Address Strobe
LDSN = lo;                                          ! Assert Lower Data Strobe
UDSN = lo;                                          ! Assert Upper Data Strobe
IABUS = A[0];                                       ! Move Jump Address From A[0]
                                                    ! To Internal Address Buffer
DBENABLE = hi;                                      ! Enable Data Bus
next;                                               ! Execute Pending Assignments

PHI1 = lo;                                          ! Phase 2
PHI2 = hi;                                          ! Of Clock Cycle 1
PC = IABUS;                                         ! Place Jump Address Into Program
                                                    ! Counter
next;

/***********************************************************************/
T = 2;                                              ! Clock Cycle 2
next;                                               ! Execute Assignment

PHI1 = hi;                                          ! Phase 1
PHI2 = lo;                                          ! Of Clock Cycle 2
while DTACKN eql hi                                 ! Wait For Memory To Place
      (                                             ! Data On The Bus
      next;                                         ! Execute Impending Assignments

      PHI1 = lo;                                    ! Phase 2
      PHI2 = hi;                                    ! Of Clock Cycle 2
      next;                                         ! Execute Assignments

      /*****************************************************************/
      T = 3;                                        ! Clock Cycle 3
      next;                                         ! Execute Assignment

      PHI1 = hi;                                    ! Phase 1
      PHI2 = lo;                                    ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                         ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                      ! On Data Bus And
      DTACKN = lo;                                  ! Asserts DTACKN(Added)
      next;                                         ! Execute Pending Assignments

      /*****************************************************************/
      T = 2                                         ! Return To Phase 2
                                                    ! Of Clock Cycle 2
      );
      next;                                         ! Execute Impending Assignments

/***********************************************************************/
T = 3;                                              ! Clock Cycle 3
```

```
next;                                        ! Execute Assignment

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 3
EXDBUF = DBUS;                               ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
next;                                        ! Execute Pending Assignments

/*******************************************************************/
T = 4;                                       ! Clock Cycle 4
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1
PHI2 = lo;                                   ! Of Clock Cycle 4
next;
PFR = EXDBUF;                                ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 4
ASN = hi;                                    ! Deactivate Address Strobe
LDSN = hi;                                   ! Deactivate Lower Data Strobe
UDSN = hi;                                   ! Deactivate Upper Data Strobe
DTACKN = hi;                                 ! Deactivate Data Transfer
                                             ! Acknowledge(Added)
next;
/*******************************************************************/
T = 5;                                       ! Clock Cycle 5
next;                                        ! Execute Previous Assignment

PHI1 = hi;                                   ! Phase 1 Of
PHI2 = lo;                                   ! Clock Cycle 5
RW = hi;                                     ! Memory Read
ADENABLE = lo;                               ! Disable Address Bus Buffer
DBENABLE = lo;                               ! Disable Data Bus Buffer
IABUS = PC;                                  ! Place PC On Internal Address
                                             ! Bus
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2 Of
PHI2 = hi;                                   ! Clock Cycle 5
ADENABLE = hi;                               ! Enable Address Bus Buffer
FCMODE = SRMODE;                             ! User Mode
FCSPACE = 2;                                 ! Accessing Program
EXABUF = IABUS;                              ! Gate Internal Address Bus
next;                                        ! Into External Address Buffer
ABUS = EXABUF;                               ! Address Placed On Bus(Added)
next;                                        ! Execute Pending Assignments

/*******************************************************************/
```

```
T = 6;                                 ! Clock Cycle 6
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 6
ASN = lo;                              ! Assert Address Strobe
LDSN = lo;                             ! Assert Lower Data Strobe
UDSN = lo;                             ! Assert Upper Data Strobe
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 6
next;                                  ! Execute Pending Assignments

/***************************************************************/
T = 7;                                 ! Clock Cycle 7
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 7
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 7
      next;                            ! Execute Assignments

      /***********************************************************/
      T = 8;                           ! Clock Cycle 8
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /***********************************************************/
      T = 7                           ! Return To Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments

/***************************************************************/
T = 8;                                 ! Clock Cycle 8
next;                                  ! Execute Assignment

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 8
EXLBUF = DBUS;                         ! Instruction On Data Bus
```

```
                                              ! Is Placed In External Data
                                              ! Bus Buffer
        next;                                 ! Execute Pending Assignments

        /*****************************************************************/
        T = 9;                                ! Clock Cycle 9
        next;                                 ! Execute Assignment

        PHI1 = hi;                            ! Phase 1
        PHI2 = lo;                            ! Of Clock Cycle 9
        PFR = EXDBUF;                         ! The Contents Of The External
                                              ! Data Bus Buffer Are Placed
                                              ! In Prefetch Register
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2
        PHI2 = hi;                            ! Of Clock Cycle 9
        ASN = hi;                             ! Deactivate Address Strobe
        LDSN = hi;                            ! Deactivate Lower Data Strobe
        UDSN = hi;                            ! Deactivate Upper Data Strobe
        PC = PC + 2;                          ! Increment Program Counter
        IR = PFR;                             ! Place Contents Of Prefetch
                                              ! Register Into Instruction
                                              ! Register
        DTACKN = hi;                          ! Deactivate Data Transfer
                                              ! Acknowledge(Added)
        next;                                 ! Execute Pending Assignments
        T = 0                                 ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             032001: move     ! MOVE.W D1,D2
                             047320: jmp      ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
     (
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
            (
            decode_execute_prefetch
            )
     )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W D1,(A1) INSTRUCTION          */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                      Structure Declarations                           */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                 m68000 Programming Registers                          */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/*************************************************************************/
/*                                                                       */
/*                 Temporary Internal Registers                          */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,                 ! Temporary Address Storage For
                              ! Exception Handler Routine
T<7:0>,                       ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/*********************************************************************/
/*                                                                 */
/*      Model transformation modifications:                        */
/*                                                                 */
/*            1) CDL decoder structure nonexistent in ISP' and un-  */
/*      necessary for model. Eliminated.                           */
/*            2) Multi-phase clock structure nonexistent in ISP'.   */
/*      Operations on registers will provide its equivalent.       */
/*            3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                      */
/*            4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                */
/*            5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing  */
/*      MC68000 assembler and linker/loader models. The memory was  */
/*      also reduced from 8 Mwords to 32 Kbytes.                   */
/*                                                                 */
/*********************************************************************/

IABUS<31:0>,                  ! Internal Address Bus
IDBUS<31:0>,                  ! Internal Data Bus
twait<7:0>,                   ! Wait Cycle Counter
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                 */
/*                External Address and Data Bus                   */
/*                                                                 */
/*********************************************************************/

DBUS<15:0>,                   ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/***********************************************************************/
/*                                                                     */
/*                      Register Subfields                             */
/*                                                                     */
/***********************************************************************/

PCADDR      = PC<23:0>,      ! Program Counter Address Field
SRTRACE     = SR<15>,        ! Trace Bit
SRMODE      = SR<13>,        ! Mode Selection Bit
SRCARRY     = SR<0>,         ! Carry Bit
SROVER      = SR<1>,         ! Overflow Bit
SRZERO      = SR<2>,         ! Zero Bit
SRNEG       = SR<3>,         ! Negative Bit
SREX        = SR<4>,         ! Extend Bit
SRMASK      = SR<10:8>,      ! Interrupt Mask
FCSPACE     = FC<1:0>,       ! Memory Access Address Space
FCMODE      = FC<2>,         ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,      ! PC Low Word
PCHI        = PC<31:16>,     ! PC High Word
D0LWORD     = D[0]<15:0>,    ! D[0] Low Word
D1LWORD     = D[1]<15:0>,    ! D[1] Low Word
D2LWORD     = D[2]<15:0>,    ! D[2] Low Word
D3LWORD     = D[3]<15:0>,    ! D[3] Low Word
D4LWORD     = D[4]<15:0>,    ! D[4] Low Word
D5LWORD     = D[5]<15:0>,    ! D[5] Low Word
D6LWORD     = D[6]<15:0>,    ! D[6] Low Word
D7LWORD     = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/***********************************************************************/
/*                                                                     */
/*                      16K 16-Bit Word Internal Memory                */
/*                                                                     */
/***********************************************************************/

M[0:32767]<7:0>;

macro

/***********************************************************************/
/*                                                                     */
/*                     Logic Level Macros                              */
/*                                                                       */
/*************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer AcKnowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[4110] = 0xff;                 ! Place Memory Locations Following The
        M[4111] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*************************************************************/
        /*                                                           */
        /*     Routine Initialization Per Hamby and Guillory         */
        /*                                                           */
        /*************************************************************/
        SRMODE = lo;                    ! Set Status Register To User Mode
        D[1] = 0x55555555;              ! Place Hex 55555555 Into D[1]
        A[0] = 0x1000;                  ! Place Hex 1000 Into A[0]
        A[1] = 0x2000;                  ! Store Data At Hex 2000
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/*************************************************************************/
/*                                                                       */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary     */
/*  to retrieve modeled instructions for simulation and analysis. It    */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /******************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /******************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /******************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
```

```
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 2
      next;                            ! Execute Assignments

      /*********************************************************/
      T = 3;                           ! Clock Cycle 3
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /*********************************************************/
      T = 2                            ! Return To  Phase 2
                                       ! Of Clock Cycle 2
      );
      next;                            ! Execute Impending Assignments

/**************************************************************/
T = 3;                                 ! Clock Cycle 3
next;                                  ! Execute Assignment

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 3
EXDBUF = DBUS;                         ! Instruction On Data Bus
                                       ! Is Placed In External Data
                                       ! Bus Buffer
next;                                  ! Execute Pending Assignments

/**************************************************************/
T = 4;                                 ! Clock Cycle 4
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 4
PFR = EXDBUF;                          ! The Contents Of The External
                                       ! Data Bus Buffer Are Placed
                                       ! In Prefetch Register
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 4
ASN = hi;                              ! Deactivate Address Strobe
LDSN = hi;                             ! Deactivate Lower Data Strobe
UDSN = hi;                             ! Deactivate Upper Data Strobe
IR = PFR;                              ! Contents Of Prefetch Register
```

```
                                          ! Are Placed Into Instruction
                                          ! Register
        DTACKN = hi;                      ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
        PC = PC + 2;                      ! Increment Program Counter
        next;                             ! Execute Pending Assignments
        T = 0                             ! Reset Clock Cycle Counter
        )


move :=                                   ! MOVE.W D1,(A1)
        (

        /*****************************************************************/

        PHI1 = hi;                        ! Phase 1 Of
        PHI2 = lo;                        ! Clock Cycle 0
        DBUS = 0xffff;                    ! Place Data Bus In High Impedance
        RW = hi;                          ! memory Read
        ADENABLE = lo;                    ! Disable Address Bus Buffer
        DBENABLE = lo;                    ! Disable Data Bus Buffer
        IABUS = PC;                       ! Place PC On Internal Address
                                          ! Bus
        next;                             ! Execute Pending Assignments

        PHI1 = lo;                        ! Phase 2 Of
        PHI2 = hi;                        ! Clock Cycle 0
        ADENABLE = hi;                    ! Enable Address Bus Buffer
        EXABUF = IABUS;                   ! Gate Internal Address Bus
                                          ! Into External Address Buffer
        FCMODE = SRMODE;                  ! User Mode
        FCSPACE = 2;                      ! Accessing Program
        next;                             ! Execute Impending Assignments
        ABUS = EXABUF;                    ! Address Placed On Bus(Added)
        next;                             ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                            ! Clock Cycle 1
        next;                             ! Execute Assignment

        PHI1 = hi;                        ! Phase 1 Of
        PHI2 = lo;                        ! Clock Cycle 1
        ASN = lo;                         ! Assert Address Strobe
        LDSN = lo;                        ! Assert Lower Data Strobe
        UDSN = lo;                        ! Assert Upper Data Strobe
        DBENABLE = hi;                    ! Enable Data Bus
        next;                             ! Execute Pending Assignments

        PHI1 = lo;                        ! Phase 2
        PHI2 = hi;                        ! Of Clock Cycle 1
        next;                             ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBUS = 0xffff;                          ! Data Bus Returned To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
IDBUS = D1LWORD;                        ! Place Low Word from D[1] On
                                        ! Internal Data Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                         ! Place Contents Of Internal
                                        ! Data Bus Into External Data Buffer
SRCARRY = lo;                           ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                   ! Execute Pending Assignments
```

```
PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
if EXDBUF eql 0                             ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                              ! Place Data On External Data Bus
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

/**********************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
if EXDBUF<15>                               ! Set Negative Condition Bit
   SRNEG = hi;                              ! If Needed
UDSN = lo;                                  ! Activate Upper And
LDSN = lo;                                  ! Lower Data Strobes
twait = 0;                                  ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      twait = twait + 1;                    ! Increment Wait Cycle
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /***************************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      if twait eql 2                        ! Memory Responds After 2 Cycles
      (
      M[ABUS] = DBUS<15:8>;                 ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;              ! In Memory
      DTACKN = lo                          ! Asserts DTACKN(Added)
      );
      next;                                 ! Execute Pending Assignments

      /***************************************************************/
      T = 7                                ! Return To Phase 2
                                           ! Of Clock Cycle 7
      );
      next;                                ! Execute Impending Assignments

/**********************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment
```

```
        PHI1 = lo;                            ! Phase 2
        PHI2 = hi;                            ! Of Clock Cycle 8
        next;                                 ! Execute Pending Assignments

        /***************************************************************/
        T = 9;                                ! Clock Cycle 9
        next;                                 ! Execute Assignment

        PHI1 = hi;                            ! Phase 1
        PHI2 = lo;                            ! Of Clock Cycle 9
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2
        PHI2 = hi;                            ! Of Clock Cycle 9
        ASN = hi;                             ! Deactivate Address Strobe
        LDSN = hi;                            ! Deactivate Lower Data Strobe
        UDSN = hi;                            ! Deactivate Upper Data Strobe
        PC = PC + 2;                          ! Increment Program Counter
        IR = PFR;                             ! Place Contents Of Prefetch
                                              ! Register Into Instruction
                                              ! Register
        DTACKN = hi;                          ! Deactivate Data Transfer
                                              ! Acknowledge(Added)
        next;                                 ! Execute Pending Assignments
        T = 0
        )


  Jmp :=                                      ! JMP (A0)
        (

        /***************************************************************/

        PHI1 = hi;                            ! Phase 1 Of
        PHI2 = lo;                            ! Clock Cycle 0
        DBUS = 0xffff;                        ! Place Data Bus In A High Impedance
        RW = hi;                              ! Memory Read
        ADENABLE = lo;                        ! Disable Address Bus Buffer
        DBENABLE = lo;                        ! Disable Data Bus Buffer
        IABUS = PC;                           ! Place PC On Internal Address
                                              ! Bus
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2 Of
        PHI2 = hi;                            ! Clock Cycle 0
        ADENABLE = hi;                        ! Enable Address Bus Buffer
        EXABUF = IABUS;                       ! Gate Internal Address Bus
                                              ! Into External Address Buffer
        FCMODE = SRMODE;                      ! User Mode
        FCSPACE = 2;                          ! Accessing Program
        next;                                 ! Execute Pending Assignments
```

```
ABUS = EXABUF;                              ! Address Placed On Bus(Added)
next;                                       ! Execute Pending Assignments

/*************************************************************/
T = 1;                                      ! Clock Cycle 1
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 1
ASN = lo;                                   ! Assert Address Strobe
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
IABUS = A[0];                               ! Move Jump Address From A[0]
                                            ! To Internal Address Buffer
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 1
PC = IABUS;                                 ! Place Jump Address Into Program
                                            ! Counter
next;

/*************************************************************/
T = 2;                                      ! Clock Cycle 2
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 2
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 2
      next;                                 ! Execute Assignments

      /*******************************************************/
      T = 3;                                ! Clock Cycle 3
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /*******************************************************/
      T = 2                                 ! Return To Phase 2
                                            ! Of Clock Cycle 2
      );
      next;                                 ! Execute Impending Assignments
```

```
/**************************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/**************************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/**************************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
```

```
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                      ! Clock Cycle 6
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 6
ASN = lo;                                   ! Assert Address Strobe
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /**********************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /**********************************************************/
      T = 7                               ! Return To Phase 2
                                          ! Of Clock Cycle 7
      );
      next;                               ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment
```

```
        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 8
        EXDBUF = DBUS;                          ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
        next;                                   ! Execute Pending Assignments

        /****************************************************************/
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                          (
                          case IR
                               031201: move     ! MOVE.W D1,(A1)
                               047320: jmp      ! JMP (A0) If IR = Octal Value
                          esac
                          )

    main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
              (
              decode_execute_prefetch
              )
        )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.L D1,A1 INSTRUCTION           */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                        Structure Declarations                         */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                   M68000 Programming Registers                        */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,              ! 8 Data Registers
A[0:6]<31:0>,              ! 7 Address Registers
UA7<31:0>,                 ! User Stack Pointer
SA7<31:0>,                 ! System Stack Pointer
PC<31:0>,                  ! Program Counter
SR<15:0>,                  ! Status Register

/*************************************************************************/
/*                                                                       */
/*                   Temporary Internal Registers                        */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                 ! Prefetch Register
IR<15:0>,                  ! Instruction Register
FC<2:0>,                   ! Function Code Register
EXDBUF<15:0>,              ! External Data Bus Buffer Register
EXABUF<23:1>,              ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,             ! ALU Buffer 1
ALUBUF2<31:0>,             ! ALU Buffer 2
DTEMP<15:0>,               ! Temporary Data Storage
DISREG<31:0>,              ! Temporary Displacement Storage
SRTEMP<15:0>,              ! Temporary Status Register Storage
                           ! (Exception Processing)
IRTEMP<15:0>,              ! Temporary Instruction Register Storage
                           ! (Exception Processing)
TEMPADR<31:0>,             ! Temporary Cycle Address Storage
                           ! (Exception Processing)
ACTYPE<15:0>,              ! Temporary Access Type Storage
                           ! (Exception Processing)
VECADR<23:0>,              ! Temporary Vector Address Storage
                           ! (Exception Processing)
```

11

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                          */
/*                                                                    */
/*           1) CDL decoder structure nonexistent in ISP' and un-     */
/*      necessary for model. Eliminated.                              */
/*           2) Multi-phase clock structure nonexistent in ISP'.      */
/*      Operations on registers will provide its equivalent.          */
/*           3) Switch structure nonexistent in ISP'. Operation on a  */
/*      register will provide its equivalent.                         */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*           5) The memory word length was reduced from 16 to 8 bit   */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                        */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                  ! External Data Bus
ABUS<23:1>;                  ! External Address Bus(changed)
```

```
format

/*************************************************************************/
/*                                                                       */
/*                        Register Subfields                             */
/*                                                                       */
/*************************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/*************************************************************************/
/*                                                                       */
/*                  16K 16-Bit Word Internal Memory                      */
/*                                                                       */
/*************************************************************************/

M[0:32767]<7:0>;

macro

/*************************************************************************/
/*                                                                       */
/*                       Logic Level Macros                              */
/*                                                                       */
```

```
/****************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/****************************************************************************/
/*                                                                          */
/*  Power On and Initialization. This process was not modeled but is       */
/*  added to initialize signals and registers.                              */
/*                                                                          */
/****************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100a] = 0xff;                ! Place Memory Locations Following The
        M[0x100b] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /***************************************************************/
        /*                                                             */
        /*     Routine Initialization Per Hamby and Guillory           */
        /*                                                             */
        /***************************************************************/
        D[1] = 0x55555555;              ! Place Hex 55555555 Into D[1]
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/****************************************************************************/
/*                                                                          */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary        */
/*  to retrieve modeled instructions for simulation and analysis. It        */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory      */
/*  on page VI-15 of their thesis.                                          */
/*                                                                    */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /***************************************************************/

      PHI1 = hi;                             ! Phase 1 Of
      PHI2 = lo;                             ! Clock Cycle 0
      RW = hi;                               ! Memory Read
      ADENABLE = lo;                         ! Disable Address Bus Buffer
      DBENABLE = lo;                         ! Disable Data Bus Buffer
      IABUS = PC;                            ! Place PC On Internal Address
                                             ! Bus
      next;                                  ! Execute Pending Assignments

      PHI1 = lo;                             ! Phase 2 Of
      PHI2 = hi;                             ! Clock Cycle 0
      ADENABLE = hi;                         ! Enable Address Bus Buffer
      EXABUF = IABUS;                        ! Gate Internal Address Bus
                                             ! Into External Address Buffer
      FCMODE = SRMODE;                       ! User Mode
      FCSPACE = 2;                           ! Accessing Program
      next;                                  ! Execute Impending Assignments
      ABUS = EXABUF;                         ! Address Placed On Bus(Added)
      next;                                  ! Execute Pending Assignments

      /***************************************************************/
      T = 1;                                 ! Clock Cycle 1
      next;                                  ! Execute Assignment

      PHI1 = hi;                             ! Phase 1 Of
      PHI2 = lo;                             ! Clock Cycle 1
      ASN = lo;                              ! Assert Address Strobe
      LDSN = lo;                             ! Assert Lower Data Strobe
      UDSN = lo;                             ! Assert Upper Data Strobe
      DBENABLE = hi;                         ! Enable Data Bus
      next;                                  ! Execute Pending Assignments

      PHI1 = lo;                             ! Phase 2
      PHI2 = hi;                             ! Of Clock Cycle 1
      next;                                  ! Execute Pending Assignments

      /***************************************************************/
      T = 2;                                 ! Clock Cycle 2
      next;                                  ! Execute Assignment

      PHI1 = hi;                             ! Phase 1
      PHI2 = lo;                             ! Of Clock Cycle 2
      while DTACKN eql hi                    ! Wait For Memory To Place
            (                                ! Data On The Bus
            next;                            ! Execute Impending Assignments
```

```
          PHI1 = lo;                        ! Phase 2
          PHI2 = hi;                        ! Of Clock Cycle 2
          next;                             ! Execute Assignments

          /*******************************************************/
          T = 3;                            ! Clock Cycle 3
          next;                             ! Execute Assignment

          PHI1 = hi;                        ! Phase 1
          PHI2 = lo;                        ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
          DTACKN = lo;                      ! Asserts DTACKN(Added)
          next;                             ! Execute Pending Assignments

          /*******************************************************/
          T = 2                             ! Return To  Phase 2
                                            ! Of Clock Cycle 2
          );
          next;                             ! Execute Impending Assignments

      /***********************************************************/
      T = 3;                                ! Clock Cycle 3
      next;                                 ! Execute Assignment

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 3
      EXDBUF = DBUS;                        ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
      next;                                 ! Execute Pending Assignments

      /***********************************************************/
      T = 4;                                ! Clock Cycle 4
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 4
      PFR = EXDBUF;                         ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 4
      ASN = hi;                             ! Deactivate Address Strobe
      LDSN = hi;                            ! Deactivate Lower Data Strobe
      UDSN = hi;                            ! Deactivate Upper Data Strobe
      IR = PFR;                             ! Contents Of Prefetch Register
                                            ! Are Placed Into Instruction
                                            ! Register
      DTACKN = hi;                          ! Deactivate Data Transfer(Added)
```

```
                                          ! Acknowledge
     PC = PC + 4;                         ! Increment Program Counter
     next;                                ! Execute Pending Assignments
     T = 0                                ! Reset Clock Cycle Counter
     )

andi :=                                   ! AND.W #$DFFF,SR
     (
     SRMODE = lo;                         ! Effect Of Instruction
     IR<15:8> = M[PC];                    ! Prefetch Next Instruction
     IR<7:0> = M[PC + 1];
     next;                                ! Is To Set Status Register
       PC = PC + 2;                         ! Increment Program Counter
     T = 5;                               ! Supervisor Bit To User
     next;                                ! Mode
     T = 0                                ! Requires 6 Clock Cycles
     )

move :=                                   ! MOVE.L D1,A1
     (

     /*****************************************************************/

     PHI1 = hi;                           ! Phase 1 Of
     PHI2 = lo;                           ! Clock Cycle 0
     ADENABLE = lo;                       ! Disable Address Bus
     DBENABLE = lo;                       ! Disable Data Bus
     DBUS = 0xffff;                       ! Place Data Bus In High Impedance
     RW = hi;                             ! Memory Read
     IABUS = PC;                          ! Place PC On Internal Address
                                          ! Bus
     IDBUS = D[1];                      ! Place Data From D[1] Onto
                                          ! Internal Data Bus
     next;                                ! Execute Pending Assignments

     PHI1 = lo;                           ! Phase 2 Of
     PHI2 = hi;                           ! Clock Cycle 0
     ADENABLE = hi;                       ! Enable Address Bus Buffer
     EXABUF = IABUS;                      ! Gate Internal Address Bus
                                          ! Into External Address Buffer
     FCMODE = SRMODE;                     ! User Mode
     FCSPACE = 2;                         ! Accessing Program
     SRCARRY = lo;                        ! Clear Status Register Carry Bit
     SROVER = lo;                         ! Clear Status Register Overflow Bit
     SRZERO = lo;                         ! Clear Status Register Zero Bit
     SRNEG  = lo;                         ! Clear Status Register Negative Bit
     A[1] = IDBUS;                      ! Place Data From Internal Data Bus
                                          ! Into A[1]
     next;                                ! Execute Impending Assignments
     ABUS = EXABUF;                       ! Address Placed On Bus(Added)
     next;                                ! Execute Pending Assignments

     /*****************************************************************/
```

```
T = 1;                                    ! Clock Cycle 1
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 1
ASN = lo;                                 ! Assert Address Strobe
LDSN = lo;                                ! Assert Lower Data Strobe
UDSN = lo;                                ! Assert Upper Data Strobe
DBENABLE = hi;                            ! Enable Data Bus
if A[1] eql 0                           ! Set Status Register Zero Bit
   SRZERO = hi;                           ! If Moved Data Is Zero
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 1
if A[1]<31>                               ! Set Status Register Negative
   SRNEG = hi;                            ! Bit If Moved Data Is Negative
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                    ! Clock Cycle 2
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 2
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 2
      next;                               ! Execute Assignments

      /***********************************************************/
      T = 3;                              ! Clock Cycle 3
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
      DTACKN = lo;                        ! Asserts DTACKN(Added)
      next;                               ! Execute Pending Assignments

      /***********************************************************/
      T = 2                               ! Return To  Phase 2
                                          ! Of Clock Cycle 2
      );
      next;                               ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                    ! Clock Cycle 3
next;                                     ! Execute Assignment
```

```
        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 3
        EXDBUF = DBUS;                          ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
        next;                                   ! Execute Pending Assignments

        /****************************************************************/
        T = 4;                                  ! Clock Cycle 4
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 4
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    jmp :=                                      ! JMP (A0)
        (

        /****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
```

```
                                               ! Into External Address Buffer
FCMODE = SRMODE;                               ! User Mode
FCSPACE = 2;                                   ! Accessing Program
next;                                          ! Execute Pending Assignments
ABUS = EXABUF;                                 ! Address Placed On Bus(Added)
next;                                          ! Execute Pending Assignments

/****************************************************************/
T = 1;                                         ! Clock Cycle 1
next;                                          ! Execute Assignment

PHI1 = hi;                                     ! Phase 1 Of
PHI2 = lo;                                     ! Clock Cycle 1
ASN = lo;                                      ! Assert Address Strobe
LDSN = lo;                                     ! Assert Lower Data Strobe
UDSN = lo;                                     ! Assert Upper Data Strobe
IABUS = A[0];                                  ! Move Jump Address From A[0]
                                               ! To Internal Address Buffer
DBENABLE = hi;                                 ! Enable Data Bus
next;                                          ! Execute Pending Assignments

PHI1 = lo;                                     ! Phase 2
PHI2 = hi;                                     ! Of Clock Cycle 1
PC = IABUS;                                    ! Place Jump Address Into Program
                                               ! Counter
next;

/****************************************************************/
T = 2;                                         ! Clock Cycle 2
next;                                          ! Execute Assignment

PHI1 = hi;                                     ! Phase 1
PHI2 = lo;                                     ! Of Clock Cycle 2
while DTACKN eql hi                            ! Wait For Memory To Place
      (                                        ! Data On The Bus
      next;                                    ! Execute Impending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 2
      next;                                    ! Execute Assignments

      /**********************************************************/
      T = 3;                                   ! Clock Cycle 3
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                    ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                 ! On Data Bus And
      DTACKN = lo;                             ! Asserts DTACKN(Added)
      next;                                    ! Execute Pending Assignments

      /**********************************************************/
```

```
          T = 2                                 ! Return To Phase 2
                                                ! Of Clock Cycle 2
          );
          next;                                 ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                          ! Clock Cycle 3
next;                                           ! Execute Assignment

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 3
EXDBUF = DBUS;                                  ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
next;                                           ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                          ! Clock Cycle 4
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 4
next;
PFR = EXDBUF;                                   ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 4
ASN = hi;                                       ! Deactivate Address Strobe
LDSN = hi;                                      ! Deactivate Lower Data Strobe
UDSN = hi;                                      ! Deactivate Upper Data Strobe
DTACKN = hi;                                    ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                          ! Clock Cycle 5
next;                                           ! Execute Previous Assignment

PHI1 = hi;                                      ! Phase 1 Of
PHI2 = lo;                                      ! Clock Cycle 5
RW = hi;                                        ! Memory Read
ADENABLE = lo;                                  ! Disable Address Bus Buffer
DBENABLE = lo;                                  ! Disable Data Bus Buffer
IABUS = PC;                                     ! Place PC On Internal Address
                                                ! Bus
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2 Of
PHI2 = hi;                                      ! Clock Cycle 5
ADENABLE = hi;                                  ! Enable Address Bus Buffer
FCMODE = SRMODE;                                ! User Mode
```

```
FCSPACE = 2;                           ! Accessing Program
EXABUF = IABUS;                        ! Gate Internal Address Bus
next;                                  ! Into External Address Buffer
ABUS = EXABUF;                         ! Address Placed On Bus(Added)
next;                                  ! Execute Pending Assignments

/**************************************************************/
T = 6;                                 ! Clock Cycle 6
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 6
ASN = lo;                              ! Assert Address Strobe
LDSN = lo;                             ! Assert Lower Data Strobe
UDSN = lo;                             ! Assert Upper Data Strobe
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 6
next;                                  ! Execute Pending Assignments

/**************************************************************/
T = 7;                                 ! Clock Cycle 7
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 7
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 7
      next;                            ! Execute Assignments

      /*******************************************************/
      T = 8;                           ! Clock Cycle 8
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /*******************************************************/
      T = 7                           ! Return To Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments
```

```
        /*****************************************************************/
        T = 8;                                    ! Clock Cycle 8
        next;                                     ! Execute Assignment

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 8
        EXDBUF = DBUS;                            ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
        next;                                     ! Execute Pending Assignments

        /*****************************************************************/
        T = 9;                                    ! Clock Cycle 9
        next;                                     ! Execute Assignment

        PHI1 = hi;                                ! Phase 1
        PHI2 = lo;                                ! Of Clock Cycle 9
        PFR = EXDBUF;                             ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Prefetch Register
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 9
        ASN = hi;                                 ! Deactivate Address Strobe
        LDSN = hi;                                ! Deactivate Lower Data Strobe
        UDSN = hi;                                ! Deactivate Upper Data Strobe
        PC = PC + 2;                              ! Increment Program Counter
        IR = PFR;                                 ! Place Contents Of Prefetch
                                                  ! Register Into Instruction
                                                  ! Register
        DTACKN = hi;                              ! Deactivate Data Transfer
                                                  ! Acknowledge(Added)
        next;                                     ! Execute Pending Assignments
        T = 0                                     ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                              (
                              case IR
                                    0x2241: move   ! MOVE.L D1,A1
                                    0x027c: andi   ! AND.W #$DFFF,SR
                                    047320: jmp    ! JMP (A0) If IR = Octal Value
                              esac
                              )

    main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
              (
              decode_execute_prefetch
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W D1,(A1)+ INSTRUCTION         */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                       Structure Declarations                          */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                   M68000 Programming Registers                        */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,                 ! 8 Data Registers
A[0:6]<31:0>,                 ! 7 Address Registers
UA7<31:0>,                    ! User Stack Pointer
SA7<31:0>,                    ! System Stack Pointer
PC<31:0>,                     ! Program Counter
SR<15:0>,                     ! Status Register

/*************************************************************************/
/*                                                                       */
/*                   Temporary Internal Registers                        */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                    ! Prefetch Register
IR<15:0>,                     ! Instruction Register
FC<2:0>,                      ! Function Code Register
EXDBUF<15:0>,                 ! External Data Bus Buffer Register
EXABUF<23:1>,                 ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,                ! ALU Buffer 1
ALUBUF2<31:0>,                ! ALU Buffer 2
DTEMP<15:0>,                  ! Temporary Data Storage
DISREG<31:0>,                 ! Temporary Displacement Storage
SRTEMP<15:0>,                 ! Temporary Status Register Storage
                              ! (Exception Processing)
IRTEMP<15:0>,                 ! Temporary Instruction Register Storage
                              ! (Exception Processing)
TEMPADR<31:0>,                ! Temporary Cycle Address Storage
                              ! (Exception Processing)
ACTYPE<15:0>,                 ! Temporary Access Type Storage
                              ! (Exception Processing)
VECADR<23:0>,                 ! Temporary Vector Address Storage
                              ! (Exception Processing)
```

```
HANADR<31:0>,               ! Temporary Address Storage For
                            ! Exception Handler Routine
T<7:0>,                     ! Clock Cycle Counter
RESET,                   ! Reset Flip-Flop
HALT,                    ! Halt Flip-Flop
RW,                      ! Read/Write Flip-Flop
ADENABLE,                ! Address Bus Buffer Enable
DBENABLE,                ! Data Bus Buffer Enable
ASN,                     ! Address Strobe Flip-Flop
LDSN,                    ! Lower Data Strobe Flip-Flop
UDSN,                    ! Upper Data Strobe Flip-Flop
DTACKN,                  ! Data Transfer Acknowledge Flip-Flop
COUT,                    ! Carry Flip-Flop
EXCEPT,                  ! Exception Processing Flip-Flop
READY,                   ! Ready Flip-Flop

/*********************************************************************/
/*                                                                 */
/*      Model transformation modifications:                        */
/*                                                                 */
/*           1) CDL decoder structure nonexistent in ISP' and un-  */
/*      necessary for model. Eliminated.                           */
/*           2) Multi-phase clock structure nonexistent in ISP'.   */
/*      Operations on registers will provide its equivalent.       */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                      */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                */
/*           5) The memory word length was reduced from 16 to 8 bit */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing */
/*      MC68000 assembler and linker/loader models. The memory was */
/*      also reduced from 8 Mwords to 32 Kbytes.                   */
/*                                                                 */
/*********************************************************************/

IABUS<31:0>,                ! Internal Address Bus
IDBUS<31:0>,                ! Internal Data Bus
twait<4:0>,                 ! Wait State Counter
SWITCH,                  ! Power Switch
PHI1,                    ! Phase 1 Of Two-Phase Clock
PHI2;                    ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                 */
/*              External Address and Data Bus                      */
/*                                                                 */
/*********************************************************************/

DBUS<15:0>,                 ! External Data Bus
```

```
ABUS<23:1>;                      ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                      Register Subfields                            */
/*                                                                    */
/**********************************************************************/

PCADDR       = PC<23:0>,        ! Program Counter Address Field
SRTRACE      = SR<15>,          ! Trace Bit
SRMODE       = SR<13>,          ! Mode Selection Bit
SRCARRY      = SR<0>,           ! Carry Bit
SROVER       = SR<1>,           ! Overflow Bit
SRZERO       = SR<2>,           ! Zero Bit
SRNEG        = SR<3>,           ! Negative Bit
SREX         = SR<4>,           ! Extend Bit
SRMASK       = SR<10:8>,        ! Interrupt Mask
FCSPACE      = FC<1:0>,         ! Memory Access Address Space
FCMODE       = FC<2>,           ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,        ! PC Low Word
PCHI         = PC<31:16>,       ! PC High Word
D0LWORD      = D[0]<15:0>,      ! D[0] Low Word
D1LWORD      = D[1]<15:0>,      ! D[1] Low Word
D2LWORD      = D[2]<15:0>,      ! D[2] Low Word
D3LWORD      = D[3]<15:0>,      ! D[3] Low Word
D4LWORD      = D[4]<15:0>,      ! D[4] Low Word
D5LWORD      = D[5]<15:0>,      ! D[5] Low Word
D6LWORD      = D[6]<15:0>,      ! D[6] Low Word
D7LWORD      = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                  16K 16-Bit Word Internal Memory                   */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                      Logic Level Macros                            */
/*                                                                     */
/***********************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/***********************************************************************/
/*                                                                     */
/*  Power On and Initialization. This process was not modeled but is   */
/*  added to initialize signals and registers.                         */
/*                                                                     */
/***********************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                        ! Turn Power On
        next;                               ! Execute Assignment
        READY = lo;                         ! System Not Ready
        RESET = lo;                         ! Assert Reset For
        delay(100);                         ! 100 Miliseconds(Active Low)
        RESET = hi;                         ! Deactivate Reset
        next;                               ! Execute Pending Assignments
        ASN = hi;                           ! Initialize Address Strobe
        LDSN = hi;                          ! Initialize Lower Data Strobe
        UDSN = hi;                          ! Initialize Upper Data Strobe
        DTACKN = hi;                        ! Initialize Data Transfer Acknowledge
        RW = hi;                            ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                     ! Place Data Bus In High Impedance State
        M[0x100c] = 0xff;                    ! Place Memory Locations Following The
        M[0x100d] = 0xff;                     ! JMP Instruction In A High State
        HALT = hi;                          ! Initialize Halt Flip-Flop(Active
                                            ! Low)
        T = 0;                              ! Initialize Clock Cycle Counter
        READY = hi;                         ! System Ready
        /**************************************************************/
        /*                                                            */
        /*      Routine Initialization Per Hamby and Guillory         */
        /*                                                            */
        /**************************************************************/
        D[1] = 0x5555;                  ! Place Hex 5555 Into D[1]
        D[2] = 0x2000;                  ! Will Be Used To Reset A[1]
        A[0] = 0x1004;                      ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;                   ! Store Data At This Address
        PC = 0x1000;                        ! Place Hex 1000 Into Program Counter
        next                                ! Execute Assignments
        )

/***********************************************************************/
/*                                                                     */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /*****************************************************************/

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 0
      RW = hi;                              ! Memory Read
      ADENABLE = lo;                        ! Disable Address Bus Buffer
      DBENABLE = lo;                        ! Disable Data Bus Buffer
      IABUS = PC;                           ! Place PC On Internal Address
                                            ! Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2 Of
      PHI2 = hi;                            ! Clock Cycle 0
      ADENABLE = hi;                        ! Enable Address Bus Buffer
      EXABUF = IABUS;                       ! Gate Internal Address Bus
                                            ! Into External Address Buffer
      FCMODE = SRMODE;                      ! User Mode
      FCSPACE = 2;                          ! Accessing Program
      next;                                 ! Execute Impending Assignments
      ABUS = EXABUF;                        ! Address Placed On Bus(Added)
      next;                                 ! Execute Pending Assignments

      /*****************************************************************/
      T = 1;                                ! Clock Cycle 1
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 1
      ASN = lo;                             ! Assert Address Strobe
      LDSN = lo;                            ! Assert Lower Data Strobe
      UDSN = lo;                            ! Assert Upper Data Strobe
      DBENABLE = hi;                        ! Enable Data Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 1
      next;                                 ! Execute Pending Assignments

      /*****************************************************************/
      T = 2;                                ! Clock Cycle 2
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 2
```

```
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 2
      next;                            ! Execute Assignments

      /**********************************************************/
      T = 3;                           ! Clock Cycle 3
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /**********************************************************/
      T = 2                            ! Return To  Phase 2
                                       ! Of Clock Cycle 2
      );
      next;                            ! Execute Impending Assignments

/******************************************************************/
T = 3;                                 ! Clock Cycle 3
next;                                  ! Execute Assignment

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 3
EXDBUF = DBUS;                         ! Instruction On Data Bus
                                       ! Is Placed In External Data
                                       ! Bus Buffer
next;                                  ! Execute Pending Assignments

/******************************************************************/
T = 4;                                 ! Clock Cycle 4
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 4
PFR = EXDBUF;                          ! The Contents Of The External
                                       ! Data Bus Buffer Are Placed
                                       ! In Prefetch Register
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 4
ASN = hi;                              ! Deactivate Address Strobe
LDSN = hi;                             ! Deactivate Lower Data Strobe
UDSN = hi;                             ! Deactivate Upper Data Strobe
IR = PFR;                              ! Contents Of Prefetch Register
```

```
                                        ! Are Placed Into Instruction
                                        ! Register
      DTACKN = hi;                      ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
      PC = PC + 4;                      ! Increment Program Counter
      next;                             ! Execute Pending Assignments
      T = 0                             ! Reset Clock Cycle Counter
      )

andi :=                                 ! AND.W #$DFFF,SR
     (
     SRMODE = lo;                       ! Effect Of Instruction
     IR<15:8> = M[PC];                  ! Prefetch Next Instruction
     IR<7:0> = M[PC + 1];
     next;                              ! Is To Set Status Register
        PC = PC + 2;                      ! Increment Program Counter
     T = 5;                             ! Supervisor Bit To User
     next;                              ! Mode
     T = 0                              ! Requires 6 Clock Cycles
     )

moveinc :=                                ! MOVE.W D1,(A1)+
      (

      /*************************************************************/

      PHI1 = hi;                        ! Phase 1 Of
      PHI2 = lo;                        ! Clock Cycle 0
      DBUS = 0xffff;                    ! Place Data Bus In High Impedance
      RW = hi;                          ! Memory Read
      ADENABLE = lo;                    ! Disable Address Bus Buffer
      ABUS = 0xffffff;                  ! Address Bus High Impedanced
      DBENABLE = lo;                    ! Disable Data Bus Buffer
      IABUS<31:1> = PC<31:1>;           ! Place PC On Internal Address
                                        ! Bus
      next;                             ! Execute Pending Assignments

      PHI1 = lo;                        ! Phase 2 Of
      PHI2 = hi;                        ! Clock Cycle 0
      ADENABLE = hi;                    ! Enable Address Bus Buffer
      EXABUF = IABUS<23:1>;             ! Gate Internal Address Bus
                                        ! Into External Address Buffer
      FCMODE = SRMODE;                  ! User Mode
      FCSPACE = 2;                      ! Accessing Program
      ABUS = IABUS<23:1>;               ! Address Placed On Bus
      next;                             ! Execute Impending Assignments

      /*************************************************************/
      T = 1;                            ! Clock Cycle 1
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1 Of
      PHI2 = lo;                        ! Clock Cycle 1
```

```
ASN = lo;                                ! Assert Address Strobe
LDSN = lo;                               ! Assert Lower Data Strobe
UDSN = lo;                               ! Assert Upper Data Strobe
DBENABLE = hi;                           ! Enable Data Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 1
next;                                    ! Execute Pending Assignments

/**************************************************************/
T = 2;                                   ! Clock Cycle 2
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 2
      next;                              ! Execute Assignments

      /*******************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*******************************************************/
      T = 2                              ! Return To  Phase 2
                                         ! Of Clock Cycle 2
      );
      next;                              ! Execute Impending Assignments

/**************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/**************************************************************/
```

```
T = 4;                                      ! Clock Cycle 4
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 4
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 4
ASN = hi;                                   ! Deactivate Address Strobe
LDSN = hi;                                  ! Deactivate Lower Data Strobe
UDSN = hi;                                  ! Deactivate Upper Data Strobe
                                            ! Are Placed Into Instruction
                                            ! Register
DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                            ! Acknowledge
next;

/****************************************************************/
T = 5;                                      ! Clock Cycle 5
next;                                       ! Execute Previous Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 5
RW = hi;                                    ! Memory Read
ADENABLE = lo;                              ! Disable Address Bus Buffer
ABUS = 0xffffff;                            ! Address Bus High Impedanced
DBUS = 0xffff;                              ! Data Bus Returned To High
                                            ! Impedance State
DBENABLE = lo;                              ! Disable Data Bus Buffer
IABUS = A[1];                               ! Place A[1] On Internal Address
                                            ! Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2 Of
PHI2 = hi;                                  ! Clock Cycle 5
ADENABLE = hi;                              ! Enable Address Bus Buffer
FCMODE = SRMODE;                            ! User Mode
FCSPACE = 1;                                ! Accessing Program
EXABUF = IABUS<23:1>;                       ! Gate Internal Address Bus
IDBUS = D1LWORD;                            ! Place Low Word from D[1] On
                                            ! Internal Data Bus
ABUS = IABUS;                               ! Place Address On Bus
next;                                       ! Into External Address Buffer

/****************************************************************/
T = 6;                                      ! Clock Cycle 6
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
```

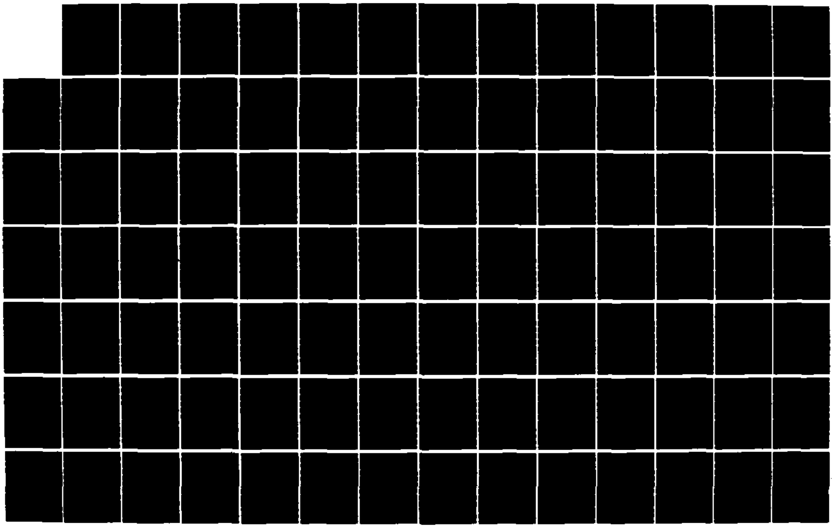
```
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                         ! Place Contents Of Internal
                                        ! Data Bus Into External Data Buffer
SRCARRY = lo;                           ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
if EXDBUF eql 0                         ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                          ! Place Data On External Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
if EXDBUF<15>                           ! Set Negative Condition Bit
   SRNEG = hi;                          ! If Needed
UDSN = lo;                              ! Activate Upper And
LDSN = lo;                              ! Lower Data Strobes
twait = 0;                              ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      twait = twait + 1;                ! Increment Wait Cycle
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 7
      next;                             ! Execute Assignments

      /*************************************************************/
      T = 8;                            ! Clock Cycle 8
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 8
      if twait eql 2                    ! Memory Responds After 2 Cycles
      (
      M[ABUS] = DBUS<15:8>;             ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;          ! In Memory
      DTACKN = lo                      ! Asserts DTACKN(Added)
      );
      next;                             ! Execute Pending Assignments
```

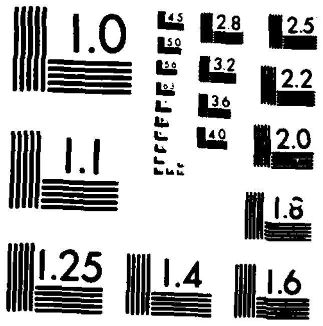
```
          /**********************************************************/
          T = 7                                   ! Return To Phase 2
                                                  ! Of Clock Cycle 7
          );
          next;                                   ! Execute Impending Assignments

      /**************************************************************/
      T = 8;                                      ! Clock Cycle 8
      next;                                       ! Execute Assignment

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 8
      next;                                       ! Execute Pending Assignments

      /**************************************************************/
      T = 9;                                      ! Clock Cycle 9
      next;                                       ! Execute Assignment

      PHI1 = hi;                                  ! Phase 1
      PHI2 = lo;                                  ! Of Clock Cycle 9
      A[1] = A[1] + 2;                            ! Increment A[1]
      next;                                       ! Execute Pending Assignments

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 9
      ASN = hi;                                   ! Deactivate Address Strobe
      LDSN = hi;                                  ! Deactivate Lower Data Strobe
      UDSN = hi;                                  ! Deactivate Upper Data Strobe
      PC = PC + 2;                                ! Increment Program Counter
      IR = PFR;                                   ! Place Contents Of Prefetch
                                                  ! Register Into Instruction
                                                  ! Register
      DTACKN = hi;                                ! Deactivate Data Transfer
                                                  ! Acknowledge(Added)
      next;                                       ! Execute Pending Assignments
      T = 0
      )

move :=                                           ! MOVE.L D2,A1
      (

      /**************************************************************/

      PHI1 = hi;                                  ! Phase 1 Of
      PHI2 = lo;                                  ! Clock Cycle 0
      ADENABLE = lo;                              ! Disable Address Bus
      DBENABLE = lo;                              ! Disable Data Bus
      DBUS = 0xffff;                              ! Place Data Bus In High Impedance
      RW = hi;                                    ! Memory Read
      IABUS = PC;                                 ! Place PC On Internal Address
                                                  ! Bus
      IDBUS = D[2];                            ! Place Data From D[2] Onto
```

```
                                              ! Internal Data Bus
     next;                                    ! Execute Pending Assignments

     PHI1 = lo;                               ! Phase 2 Of
     PHI2 = hi;                               ! Clock Cycle 0
     ADENABLE = hi;                           ! Enable Address Bus Buffer
     EXABUF = IABUS;                          ! Gate Internal Address Bus
                                              ! Into External Address Buffer
     FCMODE = SRMODE;                         ! User Mode
     FCSPACE = 2;                             ! Accessing Program
     SRCARRY = lo;                            ! Clear Status Register Carry Bit
     SROVER = lo;                             ! Clear Status Register Overflow Bit
     SRZERO = lo;                             ! Clear Status Register Zero Bit
     SRNEG  = lo;                             ! Clear Status Register Negative Bit
     A[1] = IDBUS;                          ! Place Data From Internal Data Bus
                                              ! Into A[1]
     next;                                    ! Execute Impending Assignments
     ABUS = EXABUF;                           ! Address Placed On Bus(Added)
     next;                                    ! Execute Pending Assignments

     /*******************************************************************/
     T = 1;                                   ! Clock Cycle 1
     next;                                    ! Execute Assignment

     PHI1 = hi;                               ! Phase 1 Of
     PHI2 = lo;                               ! Clock Cycle 1
     ASN = lo;                                ! Assert Address Strobe
     LDSN = lo;                               ! Assert Lower Data Strobe
     UDSN = lo;                               ! Assert Upper Data Strobe
     DBENABLE = hi;                           ! Enable Data Bus
     if A[1] eql 0                          ! Set Status Register Zero Bit
        SRZERO = hi;                          ! If Moved Data Is Zero
     next;                                    ! Execute Pending Assignments

     PHI1 = lo;                               ! Phase 2
     PHI2 = hi;                               ! Of Clock Cycle 1
     if A[1]<31>                              ! Set Status Register Negative
        SRNEG = hi;                           ! Bit If Moved Data Is Negative
     next;                                    ! Execute Pending Assignments

     /*******************************************************************/
     T = 2;                                   ! Clock Cycle 2
     next;                                    ! Execute Assignment

     PHI1 = hi;                               ! Phase 1
     PHI2 = lo;                               ! Of Clock Cycle 2
     while DTACKN eql hi                      ! Wait For Memory To Place
            (                                 ! Data On The Bus
            next;                             ! Execute Impending Assignments

            PHI1 = lo;                        ! Phase 2
            PHI2 = hi;                        ! Of Clock Cycle 2
            next;                             ! Execute Assignments
```

AD-A164 257 THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. C A BAXLEY 2/5

UNCLASSIFIED DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

```
          /*************************************************************/
          T = 3;                         ! Clock Cycle 3
          next;                          ! Execute Assignment

          PHI1 = hi;                     ! Phase 1
          PHI2 = lo;                     ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];          ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];       ! On Data Bus And
          DTACKN = lo;                   ! Asserts DTACKN(Added)
          next;                          ! Execute Pending Assignments

          /*************************************************************/
          T = 2                          ! Return To  Phase 2
                                         ! Of Clock Cycle 2
          );
          next;                          ! Execute Impending Assignments

     /******************************************************************/
     T = 3;                              ! Clock Cycle 3
     next;                               ! Execute Assignment

     PHI1 = lo;                          ! Phase 2
     PHI2 = hi;                          ! Of Clock Cycle 3
     EXDBUF = DBUS;                      ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
     next;                               ! Execute Pending Assignments

     /******************************************************************/
     T = 4;                              ! Clock Cycle 4
     next;                               ! Execute Assignment

     PHI1 = hi;                          ! Phase 1
     PHI2 = lo;                          ! Of Clock Cycle 4
     PFR = EXDBUF;                       ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
     next;                               ! Execute Pending Assignments

     PHI1 = lo;                          ! Phase 2
     PHI2 = hi;                          ! Of Clock Cycle 4
     ASN = hi;                           ! Deactivate Address Strobe
     LDSN = hi;                          ! Deactivate Lower Data Strobe
     UDSN = hi;                          ! Deactivate Upper Data Strobe
     IR = PFR;                           ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;                        ! Increment Program Counter
     next;                               ! Execute Impending Assignments
     T = 0                               ! Reset Clock Cycle Counter
```

```
        )

jmp :=                                          ! JMP (A0)
        (

        /******************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /******************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        IABUS = A[0];                           ! Move Jump Address From A[0]
                                                ! To Internal Address Buffer
        DBENABLE = hi;                          ! Enable Data Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 1
        PC = IABUS;                             ! Place Jump Address Into Program
                                                ! Counter
        next;

        /******************************************************************/
        T = 2;                                  ! Clock Cycle 2
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
```

```
PHI2 = lo;                                ! Of Clock Cycle 2
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 2
      next;                               ! Execute Assignments

      /*************************************************************/
      T = 3;                              ! Clock Cycle 3
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
      DTACKN = lo;                        ! Asserts DTACKN(Added)
      next;                               ! Execute Pending Assignments

      /*************************************************************/
      T = 2                               ! Return To Phase 2
                                          ! Of Clock Cycle 2
      );
      next;                               ! Execute Impending Assignments

/*******************************************************************/
T = 3;                                    ! Clock Cycle 3
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 3
EXDBUF = DBUS;                            ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
next;                                     ! Execute Pending Assignments

/*******************************************************************/
T = 4;                                    ! Clock Cycle 4
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 4
next;
PFR = EXDBUF;                             ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 4
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
```

```
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments
```

```
        PHI1 = lo;                         ! Phase 2
        PHI2 = hi;                         ! Of Clock Cycle 7
        next;                              ! Execute Assignments

        /*********************************************************/
        T = 8;                             ! Clock Cycle 8
        next;                              ! Execute Assignment

        PHI1 = hi;                         ! Phase 1
        PHI2 = lo;                         ! Of Clock Cycle 8
        DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
        DTACKN = lo;                       ! Asserts DTACKN(Added)
        next;                              ! Execute Pending Assignments

        /*********************************************************/
        T = 7                             ! Return To Phase 2
                                          ! Of Clock Cycle 7
        );
        next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                     ! Clock Cycle 8
next;                                      ! Execute Assignment

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 8
EXDBUF = DBUS;                             ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                     ! Clock Cycle 9
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 9
PFR = EXDBUF;                              ! The Contents Of The External
                                           ! Data Bus Buffer Are Placed
                                           ! In Prefetch Register
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 9
ASN = hi;                                  ! Deactivate Address Strobe
LDSN = hi;                                 ! Deactivate Lower Data Strobe
UDSN = hi;                                 ! Deactivate Upper Data Strobe
PC = PC + 2;                               ! Increment Program Counter
IR = PFR;                                  ! Place Contents Of Prefetch
                                           ! Register Into Instruction
                                           ! Register
DTACKN = hi;                               ! Deactivate Data Transfer
```

```
                                              ! Acknowledge(Added)
     next;                                    ! Execute Pending Assignments
     T = 0                                    ! Reset Clock Cycle Counter
     )

decode_execute_prefetch :=
                        (
                        case IR
                            0x2242: move    ! MOVE.L D2,A1
                            0x32c1: moveinc! MOVE.W D1,(A1)+
                            0x027c: andi    ! AND.W #$DFFF,SR
                            047320: jmp     ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
    (
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
         (
         decode_execute_prefetch
         )
    )
```

101

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W D1,04(A1) INSTRUCTION        */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                     Structure Declarations                            */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                  M68000 Programming Registers                         */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,              ! 8 Data Registers
A[0:6]<31:0>,              ! 7 Address Registers
UA7<31:0>,                 ! User Stack Pointer
SA7<31:0>,                 ! System Stack Pointer
PC<31:0>,                  ! Program Counter
SR<15:0>,                  ! Status Register

/*************************************************************************/
/*                                                                       */
/*                  Temporary Internal Registers                         */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                 ! Prefetch Register
IR<15:0>,                  ! Instruction Register
FC<2:0>,                   ! Function Code Register
EXDBUF<15:0>,              ! External Data Bus Buffer Register
EXABUF<23:1>,              ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,             ! ALU Buffer 1
ALUBUF2<31:0>,             ! ALU Buffer 2
DTEMP<15:0>,               ! Temporary Data Storage
DISREG<31:0>,              ! Temporary Displacement Storage
SRTEMP<15:0>,              ! Temporary Status Register Storage
                           ! (Exception Processing)
IRTEMP<15:0>,              ! Temporary Instruction Register Storage
                           ! (Exception Processing)
TEMPADR<31:0>,             ! Temporary Cycle Address Storage
                           ! (Exception Processing)
ACTYPE<15:0>,              ! Temporary Access Type Storage
                           ! (Exception Processing)
VECADR<23:0>,              ! Temporary Vector Address Storage
                           ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/*********************************************************************/
/*                                                                  */
/*      Model transformation modifications:                         */
/*                                                                  */
/*           1) CDL decoder structure nonexistent in ISP' and un-    */
/*      necessary for model. Eliminated.                            */
/*           2) Multi-phase clock structure nonexistent in ISP'.     */
/*      Operations on registers will provide its equivalent.        */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                       */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                 */
/*           5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing  */
/*      MC68000 assembler and linker/loader models. The memory was  */
/*      also reduced from 8 Mwords to 32 Kbytes.                    */
/*                                                                  */
/*********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
twait<4:0>,                  ! Wait State Counter
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                  */
/*              External Address and Data Bus                       */
/*                                                                  */
/*********************************************************************/

DBUS<15:0>,                  ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                      Register Subfields                            */
/*                                                                    */
/**********************************************************************/

PCADDR      = PC<23:0>,       ! Program Counter Address Field
SRTRACE     = SR<15>,         ! Trace Bit
SRMODE      = SR<13>,         ! Mode Selection Bit
SRCARRY     = SR<0>,          ! Carry Bit
SROVER      = SR<1>,          ! Overflow Bit
SRZERO      = SR<2>,          ! Zero Bit
SRNEG       = SR<3>,          ! Negative Bit
SREX        = SR<4>,          ! Extend Bit
SRMASK      = SR<10:8>,       ! Interrupt Mask
FCSPACE     = FC<1:0>,        ! Memory Access Address Space
FCMODE      = FC<2>,          ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,       ! PC Low Word
PCHI        = PC<31:16>,      ! PC High Word
D0LWORD     = D[0]<15:0>,     ! D[0] Low Word
D1LWORD     = D[1]<15:0>,     ! D[1] Low Word
D2LWORD     = D[2]<15:0>,     ! D[2] Low Word
D3LWORD     = D[3]<15:0>,     ! D[3] Low Word
D4LWORD     = D[4]<15:0>,     ! D[4] Low Word
D5LWORD     = D[5]<15:0>,     ! D[5] Low Word
D6LWORD     = D[6]<15:0>,     ! D[6] Low Word
D7LWORD     = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                  16K 16-Bit Word Internal Memory                   */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                     Logic Level Macros                             */
/*                                                                       */
/*************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is     */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;               ! Place Memory Locations Following The
        M[0x100f] = 0xff;                ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*     Routine Initialization Per Hamby and Guillory             */
        /*                                                               */
        /*****************************************************************/
        D[1] = 0x5555;              ! Place Hex 5555 Into D[1]
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;               ! Store Data At This Address
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/*************************************************************************/
/*                                                                       */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary     */
/*  to retrieve modeled instructions for simulation and analysis. It     */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                     */
/*                                                                     */
/***********************************************************************/

fetch_initial_instruction :=
      (

      /****************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
      while DTACKN eql hi                     ! Wait For Memory To Place
```

```
        (                                  ! Data On The Bus
        next;                              ! Execute Impending Assignments

        PHI1 = lo;                         ! Phase 2
        PHI2 = hi;                         ! Of Clock Cycle 2
        next;                              ! Execute Assignments

        /*****************************************************/
        T = 3;                             ! Clock Cycle 3
        next;                              ! Execute Assignment

        PHI1 = hi;                         ! Phase 1
        PHI2 = lo;                         ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
        DTACKN = lo;                       ! Asserts DTACKN(Added)
        next;                              ! Execute Pending Assignments

        /*****************************************************/
        T = 2                              ! Return To  Phase 2
                                           ! Of Clock Cycle 2
        );
        next;                              ! Execute Impending Assignments

    /*********************************************************/
    T = 3;                                 ! Clock Cycle 3
    next;                                  ! Execute Assignment

    PHI1 = lo;                             ! Phase 2
    PHI2 = hi;                             ! Of Clock Cycle 3
    EXDBUF = DBUS;                         ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
    next;                                  ! Execute Pending Assignments

    /*********************************************************/
    T = 4;                                 ! Clock Cycle 4
    next;                                  ! Execute Assignment

    PHI1 = hi;                             ! Phase 1
    PHI2 = lo;                             ! Of Clock Cycle 4
    PFR = EXDBUF;                          ! The Contents Of The External
                                           ! Data Bus Buffer Are Placed
                                           ! In Prefetch Register
    next;                                  ! Execute Pending Assignments

    PHI1 = lo;                             ! Phase 2
    PHI2 = hi;                             ! Of Clock Cycle 4
    ASN = hi;                              ! Deactivate Address Strobe
    LDSN = hi;                             ! Deactivate Lower Data Strobe
    UDSN = hi;                             ! Deactivate Upper Data Strobe
    IR = PFR;                              ! Contents Of Prefetch Register
                                           ! Are Placed Into Instruction
```

```
                                              ! Register
         DTACKN = hi;                         ! Deactivate Data Transfer(Added)
                                              ! Acknowledge
         PC = PC + 4;                         ! Increment Program Counter
         next;                                ! Execute Pending Assignments
         T = 0                                ! Reset Clock Cycle Counter
         )

   andi :=                                    ! AND.W #$DFFF,SR
        (
        SRMODE = lo;                          ! Effect Of Instruction
        IR<15:8> = M[PC];                     ! Prefetch Next Instruction
        IR<7:0> = M[PC + 1];
        next;                                 ! Is To Set Status Register
           PC = PC + 2;                          ! Increment Program Counter
        T = 5;                                ! Supervisor Bit To User
        next;                                 ! Mode
        T = 0                                 ! Requires 6 Clock Cycles
        )

   move :=                                    ! MOVE.W D1,4(A1) [8(A1)]
         (

         /*****************************************************************/

         PHI1 = hi;                           ! Phase 1 Of
         PHI2 = lo;                           ! Clock Cycle 0
         DBUS = 0xffff;                       ! Place Data Bus In High Impedance
         RW = hi;                             ! Memory Read
         ADENABLE = lo;                       ! Disable Address Bus Buffer
         ABUS = 0xffffff;                     ! Address Bus High Impedanced
         DBENABLE = lo;                       ! Disable Data Bus Buffer
         IABUS<31:1> = PC<31:1>;              ! Place PC On Internal Address
                                              ! Bus
         next;                                ! Execute Pending Assignments

         PHI1 = lo;                           ! Phase 2 Of
         PHI2 = hi;                           ! Clock Cycle 0
         ADENABLE = hi;                       ! Enable Address Bus Buffer
         EXABUF = IABUS<23:1>;                ! Gate Internal Address Bus
                                              ! Into External Address Buffer
         FCMODE = SRMODE;                     ! User Mode
         FCSPACE = 2;                         ! Accessing Program
         ABUS = IABUS<23:1>;                  ! Address Placed On Bus
         next;                                ! Execute Impending Assignments

         /*****************************************************************/
         T = 1;                               ! Clock Cycle 1
         next;                                ! Execute Assignment

         PHI1 = hi;                           ! Phase 1 Of
         PHI2 = lo;                           ! Clock Cycle 1
         ASN = lo;                            ! Assert Address Strobe
```

```
LDSN = lo;                                      ! Assert Lower Data Strobe
UDSN = lo;                                      ! Assert Upper Data Strobe
DBENABLE = hi;                                  ! Enable Data Bus
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 1
next;                                           ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                          ! Clock Cycle 2
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 2
while DTACKN eql hi                             ! Wait For Memory To Place
      (                                         ! Data On The Bus
      next;                                     ! Execute Impending Assignments

      PHI1 = lo;                                ! Phase 2
      PHI2 = hi;                                ! Of Clock Cycle 2
      next;                                     ! Execute Assignments

      /**********************************************************/
      T = 3;                                    ! Clock Cycle 3
      next;                                     ! Execute Assignment

      PHI1 = hi;                                ! Phase 1
      PHI2 = lo;                                ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                     ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                  ! On Data Bus And
      DTACKN = lo;                              ! Asserts DTACKN(Added)
      next;                                     ! Execute Pending Assignments

      /**********************************************************/
      T = 2                                     ! Return To  Phase 2
                                                ! Of Clock Cycle 2
      );
      next;                                     ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                          ! Clock Cycle 3
next;                                           ! Execute Assignment

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 3
EXDBUF = DBUS;                                  ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
next;                                           ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                          ! Clock Cycle 4
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
DISREG = EXDBUF sxt 32;                 ! Store Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
PC = PC + 2;                            ! Increment Program Counter
DISREG = DISREG + A[1];                ! Add Address Register To Displacement
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/*******************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBUS = 0xffff;                          ! Data Bus Returned To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Data
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                     ! Place Address On Bus
next;                                   ! Into External Address Buffer

/*******************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
UDSN = lo;                              ! Activate Upper And
LDSN = lo;                              ! Lower Data Strobes
```

```
ASN = lo;                                   ! Assert Address Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
     (                                      ! Data On The Bus
     next;                                  ! Execute Impending Assignments

     PHI1 = lo;                             ! Phase 2
     PHI2 = hi;                             ! Of Clock Cycle 7
     next;                                  ! Execute Assignments

     /***********************************************************/
     T = 8;                                 ! Clock Cycle 8
     next;                                  ! Execute Assignment

     PHI1 = hi;                             ! Phase 1
     PHI2 = lo;                             ! Of Clock Cycle 8
     DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
     DTACKN = lo;                           ! Asserts DTACKN(Added)
     next;                                  ! Execute Pending Assignments

     /***********************************************************/
     T = 7                                  ! Return To  Phase 2
                                            ! Of Clock Cycle 7
     );
     next;                                  ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 8
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                      ! Clock Cycle 9
next;                                       ! Execute Assignment
```

```
PHI1 = hi;                          ! Phase 1
PHI2 = lo;                          ! Of Clock Cycle 9
PFR = EXDBUF;                       ! The Contents Of The External
                                    ! Data Bus Buffer Are Placed
                                    ! In Prefetch Register
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2
PHI2 = hi;                          ! Of Clock Cycle 9
ASN = hi;                           ! Deactivate Address Strobe
LDSN = hi;                          ! Deactivate Lower Data Strobe
UDSN = hi;                          ! Deactivate Upper Data Strobe
DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                    ! Acknowledge
next;                               ! Execute Pending Assignments

/*************************************************************/
T = 10;                              ! Clock Cycle 10
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1 Of
PHI2 = lo;                          ! Clock Cycle 10
DBUS = 0xffff;                      ! Place Data Bus In High Impedance
RW = hi;                            ! Memory Read
ADENABLE = lo;                      ! Disable Address Bus Buffer
ABUS = 0xffffff;                    ! Address Bus High Impedanced
DBENABLE = lo;                      ! Disable Data Bus Buffer
IABUS = DISREG;                     ! Place DISREG On Internal Address
                                    ! Bus
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2 Of
PHI2 = hi;                          ! Clock Cycle 10
ADENABLE = hi;                      ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;               ! Gate Internal Address Bus
                                    ! Into External Address Buffer
IDBUS = D1LWORD;                    ! Place Low Word Of D[1] On Bus
FCMODE = SRMODE;                    ! User Mode
FCSPACE = 1;                        ! Accessing Data
ABUS = IABUS<23:1>;                 ! Address Placed On Bus
next;                               ! Execute Impending Assignments

/*************************************************************/
T = 11;                              ! Clock Cycle 11
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1 Of
PHI2 = lo;                          ! Clock Cycle 11
ASN = lo;                           ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                     ! Place Contents Of Internal
                                    ! Data Bus Into External Data Buffer
```

```
SRCARRY = lo;                               ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 11
if EXDBUF eql 0                             ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                              ! Place Data On External Data Bus
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

/*************************************************************/
T = 12;                                      ! Clock Cycle 12
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 12
if EXDBUF<15>                               ! Set Negative Condition Bit
   SRNEG = hi;                              ! If Needed
UDSN = lo;                                  ! Activate Upper And
LDSN = lo;                                  ! Lower Data Strobes
twait = 0;                                  ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
       twait = twait + 1;                   ! Increment Wait Cycle
       next;                                ! Execute Impending Assignments

       PHI1 = lo;                           ! Phase 2
       PHI2 = hi;                           ! Of Clock Cycle 12
       next;                                ! Execute Assignments

       /*****************************************************/
       T = 13;                               ! Clock Cycle 13
       next;                                ! Execute Assignment

       PHI1 = hi;                           ! Phase 1
       PHI2 = lo;                           ! Of Clock Cycle 13
       if twait eql 2                       ! Memory Responds After 2 Cycles
       (
       M[ABUS] = DBUS<15:8>;                ! Store Data From Bus
       M[ABUS + 1] = DBUS<7:0>;             ! In Memory
       DTACKN = lo                        ! Asserts DTACKN(Added)
       );
       next;                                ! Execute Pending Assignments

       /*****************************************************/
       T = 12                               ! Return To Phase 2
                                          ! Of Clock Cycle 12
       );
```

```
          next;                         ! Execute Impending Assignments

      /****************************************************************/
      T = 13;                            ! Clock Cycle 13
      next;                             ! Execute Assignment

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 13
      next;                             ! Execute Pending Assignments

      /****************************************************************/
      T = 14;                            ! Clock Cycle 14
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 14
      next;                             ! Execute Pending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 9
      ASN = hi;                         ! Deactivate Address Strobe
      LDSN = hi;                        ! Deactivate Lower Data Strobe
      UDSN = hi;                        ! Deactivate Upper Data Strobe
      PC = PC + 2;                      ! Increment Program Counter
      IR = PFR;                         ! Place Contents Of Prefetch
                                        ! Register Into Instruction
                                        ! Register
      DTACKN = hi;                      ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
      next;                             ! Execute Pending Assignments
      T = 0
      )

jmp :=                                  ! JMP (A0)
      (

      /****************************************************************/

      PHI1 = hi;                        ! Phase 1 Of
      PHI2 = lo;                        ! Clock Cycle 0
      DBUS = 0xffff;                    ! Place Data Bus In A High Impedance
      RW = hi;                          ! Memory Read
      ADENABLE = lo;                    ! Disable Address Bus Buffer
      DBENABLE = lo;                    ! Disable Data Bus Buffer
      IABUS = PC;                       ! Place PC On Internal Address
                                        ! Bus
      next;                             ! Execute Pending Assignments

      PHI1 = lo;                        ! Phase 2 Of
      PHI2 = hi;                        ! Clock Cycle 0
      ADENABLE = hi;                    ! Enable Address Bus Buffer
      EXABUF = IABUS;                   ! Gate Internal Address Bus
                                        ! Into External Address Buffer
```

```
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
next;                                   ! Execute Pending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/**************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/**************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /********************************************************/
      T = 2                             ! Return To Phase 2
```

```
                                            ! Of Clock Cycle 2
     );
     next;                                  ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                      ! Clock Cycle 4
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 4
next;
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 4
ASN = hi;                                   ! Deactivate Address Strobe
LDSN = hi;                                  ! Deactivate Lower Data Strobe
UDSN = hi;                                  ! Deactivate Upper Data Strobe
DTACKN = hi;                                ! Deactivate Data Transfer
                                            ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                      ! Clock Cycle 5
next;                                       ! Execute Previous Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 5
RW = hi;                                    ! Memory Read
ADENABLE = lo;                              ! Disable Address Bus Buffer
DBENABLE = lo;                              ! Disable Data Bus Buffer
IABUS = PC;                                 ! Place PC On Internal Address
                                            ! Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2 Of
PHI2 = hi;                                  ! Clock Cycle 5
ADENABLE = hi;                              ! Enable Address Bus Buffer
FCMODE = SRMODE;                            ! User Mode
FCSPACE = 2;                                ! Accessing Program
```

```
EXABUF = IABUS;                           ! Gate Internal Address Bus
next;                                     ! Into External Address Buffer
ABUS = EXABUF;                            ! Address Placed On Bus(Added)
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                    ! Clock Cycle 6
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 6
ASN = lo;                                 ! Assert Address Strobe
LDSN = lo;                                ! Assert Lower Data Strobe
UDSN = lo;                                ! Assert Upper Data Strobe
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 6
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                    ! Clock Cycle 7
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 7
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 7
      next;                               ! Execute Assignments

      /***********************************************************/
      T = 8;                              ! Clock Cycle 8
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
      DTACKN = lo;                        ! Asserts DTACKN(Added)
      next;                               ! Execute Pending Assignments

      /***********************************************************/
      T = 7                              ! Return To Phase 2
                                         ! Of Clock Cycle 7
      );
      next;                              ! Execute Impending Assignments

/*****************************************************************/
```

```
        T = 8;                                      ! Clock Cycle 8
        next;                                       ! Execute Assignment

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 8
        EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                    ! Is Placed In External Data
                                                    ! Bus Buffer
        next;                                       ! Execute Pending Assignments

        /******************************************************************/
        T = 9;                                      ! Clock Cycle 9
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1
        PHI2 = lo;                                  ! Of Clock Cycle 9
        PFR = EXDBUF;                               ! The Contents Of The External
                                                    ! Data Bus Buffer Are Placed
                                                    ! In Prefetch Register
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 9
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        PC = PC + 2;                                ! Increment Program Counter
        IR = PFR;                                   ! Place Contents Of Prefetch
                                                    ! Register Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer
                                                    ! Acknowledge(Added)
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             0x3341: move           ! MOVE.W D1,4(A1)  [8(A1)]
                             0x027c: andi           ! AND.W #$DFFF,SR
                             047320: jmp            ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
     (
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           (
           decode_execute_prefetch
           )
```

```
/************************************************************************/
/*                                                                      */
/*    MOTOROLA MC68000 MODEL OF THE MOVE.W D1,04(A1,D7) INSTRUCTION     */
/*                                                                      */
/************************************************************************/

/************************************************************************/
/*                                                                      */
/*                       Structure Declarations                         */
/*                                                                      */
/************************************************************************/

state

/************************************************************************/
/*                                                                      */
/*                  M68000 Programming Registers                        */
/*                                                                      */
/************************************************************************/

D[0:7]<31:0>,                ! 8 Data Registers
A[0:6]<31:0>,                ! 7 Address Registers
UA7<31:0>,                   ! User Stack Pointer
SA7<31:0>,                   ! System Stack Pointer
PC<31:0>,                    ! Program Counter
SR<15:0>,                    ! Status Register

/************************************************************************/
/*                                                                      */
/*                  Temporary Internal Registers                        */
/*                                                                      */
/************************************************************************/

PFR<15:0>,                   ! Prefetch Register
IR<15:0>,                    ! Instruction Register
FC<2:0>,                     ! Function Code Register
EXDBUF<15:0>,                ! External Data Bus Buffer Register
EXABUF<23:1>,                ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,               ! ALU Buffer 1
ALUBUF2<31:0>,               ! ALU Buffer 2
DTEMP<15:0>,                 ! Temporary Data Storage
DISREG<31:0>,                ! Temporary Displacement Storage
SRTEMP<15:0>,                ! Temporary Status Register Storage
                             ! (Exception Processing)
IRTEMP<15:0>,                ! Temporary Instruction Register Storage
                             ! (Exception Processing)
TEMPADR<31:0>,               ! Temporary Cycle Address Storage
                             ! (Exception Processing)
ACTYPE<15:0>,                ! Temporary Access Type Storage
                             ! (Exception Processing)
VECADR<23:0>,                ! Temporary Vector Address Storage
                             ! (Exception Processing)
```

```
HANADR<31:0>,                   ! Temporary Address Storage For
                                ! Exception Handler Routine
T<7:0>,                         ! Clock Cycle Counter
RESET,                      ! Reset Flip-Flop
HALT,                       ! Halt Flip-Flop
RW,                         ! Read/Write Flip-Flop
ADENABLE,                   ! Address Bus Buffer Enable
DBENABLE,                   ! Data Bus Buffer Enable
ASN,                        ! Address Strobe Flip-Flop
LDSN,                       ! Lower Data Strobe Flip-Flop
UDSN,                       ! Upper Data Strobe Flip-Flop
DTACKN,                     ! Data Transfer Acknowledge Flip-Flop
COUT,                       ! Carry Flip-Flop
EXCEPT,                     ! Exception Processing Flip-Flop
READY,                      ! Ready Flip-Flop

/*****************************************************************************/
/*                                                                         */
/*      Model transformation modifications:                                */
/*                                                                         */
/*              1) CDL decoder structure nonexistent in ISP' and un-       */
/*      necessary for model. Eliminated.                                   */
/*              2) Multi-phase clock structure nonexistent in ISP'.        */
/*      Operations on registers will provide its equivalent.               */
/*              3) Switch structure nonexistent in ISP'. Operation on a    */
/*      register will provide its equivalent.                              */
/*              4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model       */
/*      equivalency and simplicity.                                        */
/*              5) The memory word length was reduced from 16 to 8 bit     */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing         */
/*      MC68000 assembler and linker/loader models. The memory was         */
/*      also reduced from 8 Mwords to 32 Kbytes.                           */
/*                                                                         */
/*****************************************************************************/

IABUS<31:0>,                    ! Internal Address Bus
IDBUS<31:0>,                    ! Internal Data Bus
twait<4:0>,                     ! Wait State Counter
SWITCH,                     ! Power Switch
PHI1,                       ! Phase 1 Of Two-Phase Clock
PHI2;                       ! Phase 2 Of Two-Phase Clock

port

/*****************************************************************************/
/*                                                                         */
/*              External Address and Data Bus                              */
/*                                                                         */
/*****************************************************************************/

DBUS<15:0>,                     ! External Data Bus
```

```
ABUS<23:1>;                 ! External Address Bus(changed)

format

/***********************************************************************/
/*                                                                     */
/*                    Register Subfields                               */
/*                                                                     */
/***********************************************************************/

PCADDR       = PC<23:0>,      ! Program Counter Address Field
SRTRACE      = SR<15>,        ! Trace Bit
SRMODE       = SR<13>,        ! Mode Selection Bit
SRCARRY      = SR<0>,         ! Carry Bit
SROVER       = SR<1>,         ! Overflow Bit
SRZERO       = SR<2>,         ! Zero Bit
SRNEG        = SR<3>,         ! Negative Bit
SREX         = SR<4>,         ! Extend Bit
SRMASK       = SR<10:8>,      ! Interrupt Mask
FCSPACE      = FC<1:0>,       ! Memory Access Address Space
FCMODE       = FC<2>,         ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,      ! PC Low Word
PCHI         = PC<31:16>,     ! PC High Word
D0LWORD      = D[0]<15:0>,    ! D[0] Low Word
D1LWORD      = D[1]<15:0>,    ! D[1] Low Word
D2LWORD      = D[2]<15:0>,    ! D[2] Low Word
D3LWORD      = D[3]<15:0>,    ! D[3] Low Word
D4LWORD      = D[4]<15:0>,    ! D[4] Low Word
D5LWORD      = D[5]<15:0>,    ! D[5] Low Word
D6LWORD      = D[6]<15:0>,    ! D[6] Low Word
D7LWORD      = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***********************************************************************/
/*                                                                     */
/*                    16K 16-Bit Word Internal Memory                  */
/*                                                                     */
/***********************************************************************/

M[0:32767]<7:0>;

macro

/***********************************************************************/
/*                                                                     */
/*                  Logic Level Macros                                 */
```

141

```
/*                                                                   */
/*******************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*******************************************************************/
/*                                                                   */
/*  Power On and Initialization. This process was not modeled but is */
/*  added to initialize signals and registers.                       */
/*                                                                   */
/*******************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                     ! Turn Power On
        next;                            ! Execute Assignment
        READY = lo;                      ! System Not Ready
        RESET = lo;                      ! Assert Reset For
        delay(100);                      ! 100 Miliseconds(Active Low)
        RESET = hi;                      ! Deactivate Reset
        next;                            ! Execute Pending Assignments
        ASN = hi;                        ! Initialize Address Strobe
        LDSN = hi;                       ! Initialize Lower Data Strobe
        UDSN = hi;                       ! Initialize Upper Data Strobe
        DTACKN = hi;                     ! Initialize Data Transfer Acknowledge
        RW = hi;                         ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                 ! Place Memory Locations Following The
        M[0x100f] = 0xff;                  ! JMP Instruction In A High State
        HALT = hi;                       ! Initialize Halt Flip-Flop(Active
                                         ! Low)
        T = 0;                           ! Initialize Clock Cycle Counter
        READY = hi;                      ! System Ready
        /***********************************************************/
        /*                                                           */
        /*      Routine Initialization Per Hamby and Guillory        */
        /*                                                           */
        /***********************************************************/
        D[1] = 0x5555;               ! Place Hex 5555 Into D[1]
        D[7] = 0x00000006;           ! Place 6 Into D[7]
        A[0] = 0x1004;                   ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;                ! Store Data At This Address
        PC = 0x1000;                     ! Place Hex 1000 Into Program Counter
        next                             ! Execute Assignments
        )

/*******************************************************************/
/*                                                                   */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                     */
/*                                                                     */
/*********************************************************************/

fetch_initial_instruction :=
      (

      /***************************************************************/

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 0
      RW = hi;                            ! Memory Read
      ADENABLE = lo;                      ! Disable Address Bus Buffer
      DBENABLE = lo;                      ! Disable Data Bus Buffer
      IABUS = PC;                         ! Place PC On Internal Address
                                          ! Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2 Of
      PHI2 = hi;                          ! Clock Cycle 0
      ADENABLE = hi;                      ! Enable Address Bus Buffer
      EXABUF = IABUS;                     ! Gate Internal Address Bus
                                          ! Into External Address Buffer
      FCMODE = SRMODE;                    ! User Mode
      FCSPACE = 2;                        ! Accessing Program
      next;                               ! Execute Impending Assignments
      ABUS = EXABUF;                      ! Address Placed On Bus(Added)
      next;                               ! Execute Pending Assignments

      /***************************************************************/
      T = 1;                              ! Clock Cycle 1
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 1
      ASN = lo;                           ! Assert Address Strobe
      LDSN = lo;                          ! Assert Lower Data Strobe
      UDSN = lo;                          ! Assert Upper Data Strobe
      DBENABLE = hi;                      ! Enable Data Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 1
      next;                               ! Execute Pending Assignments

      /***************************************************************/
      T = 2;                              ! Clock Cycle 2
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 2
```

```
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 2
      next;                               ! Execute Assignments

      /**********************************************************/
      T = 3;                              ! Clock Cycle 3
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
      DTACKN = lo;                        ! Asserts DTACKN(Added)
      next;                               ! Execute Pending Assignments

      /**********************************************************/
      T = 2                               ! Return To  Phase 2
                                          ! Of Clock Cycle 2
      );
      next;                               ! Execute Impending Assignments

/****************************************************************/
T = 3;                                    ! Clock Cycle 3
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 3
EXDBUF = DBUS;                            ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
next;                                     ! Execute Pending Assignments

/****************************************************************/
T = 4;                                    ! Clock Cycle 4
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 4
PFR = EXDBUF;                             ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 4
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
UDSN = hi;                                ! Deactivate Upper Data Strobe
IR = PFR;                                 ! Contents Of Prefetch Register
```

```
                                        ! Are Placed Into Instruction
                                        ! Register
        DTACKN = hi;                    ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
        PC = PC + 4;                    ! Increment Program Counter
        next;                           ! Execute Pending Assignments
        T = 0                           ! Reset Clock Cycle Counter
        )

andi :=                                 ! AND.W #$DFFF,SR
      (
      SRMODE = lo;                      ! Effect Of Instruction
      IR<15:8> = M[PC];                 ! Prefetch Next Instruction
      IR<7:0> = M[PC + 1];
      next;                             ! Is To Set Status Register
         PC = PC + 2;                     ! Increment Program Counter
      T = 5;                            ! Supervisor Bit To User
      next;                             ! Mode
      T = 0                             ! Requires 6 Clock Cycles
      )

move :=                                 ! MOVE.W D1,4(A1,D7) [8(A1,D7)]
        (

        /**********************************************************/

        PHI1 = hi;                      ! Phase 1 Of
        PHI2 = lo;                      ! Clock Cycle 0
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance
        RW = hi;                        ! Memory Read
        ADENABLE = lo;                  ! Disable Address Bus Buffer
        ABUS = 0xffffff;                ! Address Bus High Impedanced
        DBENABLE = lo;                  ! Disable Data Bus Buffer
        IABUS<31:1> = PC<31:1>;         ! Place PC On Internal Address
                                        ! Bus
        next;                           ! Execute Pending Assignments

        PHI1 = lo;                      ! Phase 2 Of
        PHI2 = hi;                      ! Clock Cycle 0
        ADENABLE = hi;                  ! Enable Address Bus Buffer
        EXABUF = IABUS<23:1>;           ! Gate Internal Address Bus
                                        ! Into External Address Buffer
        FCMODE = SRMODE;                ! User Mode
        FCSPACE = 2;                    ! Accessing Program
        ABUS = IABUS<23:1>;             ! Address Placed On Bus
        next;                           ! Execute Impending Assignments

        /**********************************************************/
        T = 1;                          ! Clock Cycle 1
        next;                           ! Execute Assignment

        PHI1 = hi;                      ! Phase 1 Of
        PHI2 = lo;                      ! Clock Cycle 1
```

```
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /**********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /**********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
```

```
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
DISREG = EXDBUF<7:0> sxt 32;            ! Store Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
PC = PC + 2;                            ! Increment Program Counter
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBUS = 0xffff;                          ! Data Bus High Impedanced
DISREG = DISREG + A[1];               ! Add Address Register To Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
DISREG = DISREG + D7LWORD;              ! Add Data Register To Displacement
next;                                   ! Into External Address Buffer

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
```

```
PHI2 = lo;                              ! Clock Cycle 7
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 7
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Data
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                     ! Place Address On Bus
next;                                   ! Into External Address Buffer

/*****************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 8
UDSN = lo;                              ! Activate Upper And
LDSN = lo;                              ! Lower Data Strobes
ASN = lo;                               ! Assert Address Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                  ! Clock Cycle 9
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 9
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 9
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 10;                           ! Clock Cycle 10
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 10
```

```
          DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
          DTACKN = lo;                       ! Asserts DTACKN(Added)
          next;                              ! Execute Pending Assignments

          /*********************************************************/
          T = 9                              ! Return To  Phase 2
                                             ! Of Clock Cycle 9
          );
          next;                              ! Execute Impending Assignments

    /*************************************************************/
    T = 10;                                   ! Clock Cycle 10
    next;                                    ! Execute Assignment

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 10
    EXDBUF = DBUS;                           ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
    next;                                    ! Execute Pending Assignments

    /*************************************************************/
    T = 11;                                   ! Clock Cycle 11
    next;                                    ! Execute Assignment

    PHI1 = hi;                               ! Phase 1
    PHI2 = lo;                               ! Of Clock Cycle 11
    PFR = EXDBUF;                            ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
    next;                                    ! Execute Pending Assignments

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 11
    ASN = hi;                                ! Deactivate Address Strobe
    LDSN = hi;                               ! Deactivate Lower Data Strobe
    UDSN = hi;                               ! Deactivate Upper Data Strobe
    DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                             ! Acknowledge
    next;                                    ! Execute Pending Assignments

    /*************************************************************/
    T = 12;                                   ! Clock Cycle 12
    next;                                    ! Execute Assignment

    PHI1 = hi;                               ! Phase 1 Of
    PHI2 = lo;                               ! Clock Cycle 12
    DBUS = 0xffff;                           ! Place Data Bus In High Impedance
    RW = hi;                                 ! Memory Read
    ADENABLE = lo;                           ! Disable Address Bus Buffer
    ABUS = 0xffffff;                         ! Address Bus High Impedanced
    DBENABLE = lo;                           ! Disable Data Bus Buffer
```

```
IABUS = DISREG;                        ! Place DISREG On Internal Address
                                       ! Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2 Of
PHI2 = hi;                             ! Clock Cycle 12
ADENABLE = hi;                         ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                  ! Gate Internal Address Bus
                                       ! Into External Address Buffer
IDBUS = D1LWORD;                       ! Place Low Word Of D[1] On Bus
FCMODE = SRMODE;                       ! User Mode
FCSPACE = 1;                           ! Accessing Data
ABUS = IABUS<23:1>;                    ! Address Placed On Bus
next;                                  ! Execute Impending Assignments

/**************************************************************************/
T = 13;                                 ! Clock Cycle 13
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 13
ASN = lo;                              ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                        ! Place Contents Of Internal
                                       ! Data Bus Into External Data Buffer
SRCARRY = lo;                          ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 13
if EXDBUF eql 0                        ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                         ! Place Data On External Data Bus
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

/**************************************************************************/
T = 14;                                 ! Clock Cycle 14
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 14
if EXDBUF<15>                          ! Set Negative Condition Bit
   SRNEG = hi;                         ! If Needed
UDSN = lo;                             ! Activate Upper And
LDSN = lo;                             ! Lower Data Strobes
twait = 0;                             ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
        twait = twait + 1;                  ! Increment Wait Cycle
        next;                               ! Execute Impending Assignments

        PHI1 = lo;                          ! Phase 2
        PHI2 = hi;                          ! Of Clock Cycle 14
        next;                               ! Execute Assignments

        /*************************************************************/
        T = 15;                              ! Clock Cycle 15
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1
        PHI2 = lo;                          ! Of Clock Cycle 15
        if twait eql 2                      ! Memory Responds After 2 Cycles
        (
        M[ABUS] = DBUS<15:8>;               ! Store Data From Bus
        M[ABUS + 1] = DBUS<7:0>;            ! In Memory
        DTACKN = lo                        ! Asserts DTACKN(Added)
        );
        next;                               ! Execute Pending Assignments

        /*************************************************************/
        T = 14                              ! Return To Phase 2
                                           ! Of Clock Cycle 14
        );
        next;                              ! Execute Impending Assignments

    /******************************************************************/
    T = 15;                                  ! Clock Cycle 15
    next;                                   ! Execute Assignment

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 15
    next;                                   ! Execute Pending Assignments

    /******************************************************************/
    T = 16;                                  ! Clock Cycle 16
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 16
    next;                                   ! Execute Pending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 16
    ASN = hi;                               ! Deactivate Address Strobe
    LDSN = hi;                              ! Deactivate Lower Data Strobe
    UDSN = hi;                              ! Deactivate Upper Data Strobe
    PC = PC + 2;                            ! Increment Program Counter
    IR = PFR;                               ! Place Contents Of Prefetch
                                            ! Register Into Instruction
                                            ! Register
    DTACKN = hi;                            ! Deactivate Data Transfer
```

```
                                                 ! Acknowledge(Added)
        next;                                    ! Execute Pending Assignments
        T = 0
        )

   Jmp :=                                        ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                               ! Phase 1 Of
        PHI2 = lo;                               ! Clock Cycle 0
        DBUS = 0xffff;                           ! Place Data Bus In A High Impedance
        RW = hi;                                 ! Memory Read
        ADENABLE = lo;                           ! Disable Address Bus Buffer
        DBENABLE = lo;                           ! Disable Data Bus Buffer
        IABUS = PC;                              ! Place PC On Internal Address
                                                 ! Bus
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2 Of
        PHI2 = hi;                               ! Clock Cycle 0
        ADENABLE = hi;                           ! Enable Address Bus Buffer
        EXABUF = IABUS;                          ! Gate Internal Address Bus
                                                 ! Into External Address Buffer
        FCMODE = SRMODE;                         ! User Mode
        FCSPACE = 2;                             ! Accessing Program
        next;                                    ! Execute Pending Assignments
        ABUS = EXABUF;                           ! Address Placed On Bus(Added)
        next;                                    ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                   ! Clock Cycle 1
        next;                                    ! Execute Assignment

        PHI1 = hi;                               ! Phase 1 Of
        PHI2 = lo;                               ! Clock Cycle 1
        ASN = lo;                                ! Assert Address Strobe
        LDSN = lo;                               ! Assert Lower Data Strobe
        UDSN = lo;                               ! Assert Upper Data Strobe
        IABUS = A[0];                            ! Move Jump Address From A[0]
                                                 ! To Internal Address Buffer
        DBENABLE = hi;                           ! Enable Data Bus
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 1
        PC = IABUS;                              ! Place Jump Address Into Program
                                                 ! Counter
        next;

        /*****************************************************************/
        T = 2;                                   ! Clock Cycle 2
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*****************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*****************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
```

```
PHI2 = hi;                               ! Of Clock Cycle 4
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
UDSN = hi;                               ! Deactivate Upper Data Strobe
DTACKN = hi;                             ! Deactivate Data Transfer
                                         ! Acknowledge(Added)
next;
/**************************************************************************/
T = 5;                                   ! Clock Cycle 5
next;                                    ! Execute Previous Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 5
RW = hi;                                 ! Memory Read
ADENABLE = lo;                           ! Disable Address Bus Buffer
DBENABLE = lo;                           ! Disable Data Bus Buffer
IABUS = PC;                              ! Place PC On Internal Address
                                         ! Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2 Of
PHI2 = hi;                               ! Clock Cycle 5
ADENABLE = hi;                           ! Enable Address Bus Buffer
FCMODE = SRMODE;                         ! User Mode
FCSPACE = 2;                             ! Accessing Program
EXABUF = IABUS;                          ! Gate Internal Address Bus
next;                                    ! Into External Address Buffer
ABUS = EXABUF;                           ! Address Placed On Bus(Added)
next;                                    ! Execute Pending Assignments

/**************************************************************************/
T = 6;                                   ! Clock Cycle 6
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 6
ASN = lo;                                ! Assert Address Strobe
LDSN = lo;                               ! Assert Lower Data Strobe
UDSN = lo;                               ! Assert Upper Data Strobe
DBENABLE = hi;                           ! Enable Data Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 6
next;                                    ! Execute Pending Assignments

/**************************************************************************/
T = 7;                                   ! Clock Cycle 7
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 7
while DTACKN eql hi                      ! Wait For Memory To Place
```

```
        (                                   ! Data On The Bus
        next;                               ! Execute Impending Assignments

        PHI1 = lo;                          ! Phase 2
        PHI2 = hi;                          ! Of Clock Cycle 7
        next;                               ! Execute Assignments

        /*************************************************************/
        T = 8;                              ! Clock Cycle 8
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1
        PHI2 = lo;                          ! Of Clock Cycle 8
        DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
        DTACKN = lo;                        ! Asserts DTACKN(Added)
        next;                               ! Execute Pending Assignments

        /*************************************************************/
        T = 7                              ! Return To Phase 2
                                           ! Of Clock Cycle 7
        );
        next;                              ! Execute Impending Assignments

    /****************************************************************/
    T = 8;                                  ! Clock Cycle 8
    next;                                   ! Execute Assignment

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 8
    EXDBUF = DBUS;                          ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
    next;                                   ! Execute Pending Assignments

    /****************************************************************/
    T = 9;                                  ! Clock Cycle 9
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 9
    PFR = EXDBUF;                           ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
    next;                                   ! Execute Pending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 9
    ASN = hi;                               ! Deactivate Address Strobe
    LDSN = hi;                              ! Deactivate Lower Data Strobe
    UDSN = hi;                              ! Deactivate Upper Data Strobe
    PC = PC + 2;                            ! Increment Program Counter
    IR = PFR;                               ! Place Contents Of Prefetch
```

```
                                                    ! Register Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer
                                                    ! AcKnowledge(Added)
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

   decode_execute_prefetch :=
                              (
                              case IR
                                   0x3381: move     ! MOVE.W D1,4(A1,D7) [8(A1,D7)]
                                   0x027c: andi     ! AND.W #$DFFF,SR
                                   047320: jmp      ! JMP (A0) If IR = Octal Value
                              esac
                              )

   main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
             (
             decode_execute_prefetch
             )
        )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W D1,$2004 INSTRUCTION         */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                      Structure Declarations                           */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                  M68000 Programming Registers                         */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,                   ! 8 Data Registers
A[0:6]<31:0>,                   ! 7 Address Registers
UA7<31:0>,                      ! User Stack Pointer
SA7<31:0>,                      ! System Stack Pointer
PC<31:0>,                       ! Program Counter
SR<15:0>,                       ! Status Register

/*************************************************************************/
/*                                                                       */
/*                  Temporary Internal Registers                         */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                      ! Prefetch Register
IR<15:0>,                       ! Instruction Register
FC<2:0>,                        ! Function Code Register
EXDBUF<15:0>,                   ! External Data Bus Buffer Register
EXABUF<23:1>,                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,                  ! ALU Buffer 1
ALUBUF2<31:0>,                  ! ALU Buffer 2
DTEMP<15:0>,                    ! Temporary Data Storage
DISREG<31:0>,                   ! Temporary Displacement Storage
SRTEMP<15:0>,                   ! Temporary Status Register Storage
                                ! (Exception Processing)
IRTEMP<15:0>,                   ! Temporary Instruction Register Storage
                                ! (Exception Processing)
TEMPADR<31:0>,                  ! Temporary Cycle Address Storage
                                ! (Exception Processing)
ACTYPE<15:0>,                   ! Temporary Access Type Storage
                                ! (Exception Processing)
VECADR<23:0>,                   ! Temporary Vector Address Storage
                                ! (Exception Processing)
```

```
HANADR<31:0>,              ! Temporary Address Storage For
                           ! Exception Handler Routine
T<7:0>,                    ! Clock Cycle Counter
RESET,                   ! Reset Flip-Flop
HALT,                    ! Halt Flip-Flop
RW,                      ! Read/Write Flip-Flop
ADENABLE,                ! Address Bus Buffer Enable
DBENABLE,                ! Data Bus Buffer Enable
ASN,                     ! Address Strobe Flip-Flop
LDSN,                    ! Lower Data Strobe Flip-Flop
UDSN,                    ! Upper Data Strobe Flip-Flop
DTACKN,                  ! Data Transfer AcKnowledge Flip-Flop
COUT,                    ! Carry Flip-Flop
EXCEPT,                  ! Exception Processing Flip-Flop
READY,                   ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                           */
/*                                                                    */
/*           1) CDL decoder structure nonexistent in ISP' and un-     */
/*      necessary for model. Eliminated.                              */
/*           2) Multi-phase clock structure nonexistent in ISP'.      */
/*      Operations on registers will provide its equivalent.          */
/*           3) Switch structure nonexistent in ISP'. Operation on a  */
/*      register will provide its equivalent.                         */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*           5) The memory word length was reduced from 16 to 8 bit   */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,               ! Internal Address Bus
IDBUS<31:0>,               ! Internal Data Bus
twait<4:0>,                ! Wait State Counter
SWITCH,                  ! Power Switch
PHI1,                    ! Phase 1 Of Two-Phase Clock
PHI2;                    ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                ! External Data Bus
```

```
ABUS<23:1>;                     ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                    Register Subfields                              */
/*                                                                    */
/**********************************************************************/

PCADDR      = PC<23:0>,        ! Program Counter Address Field
SRTRACE     = SR<15>,          ! Trace Bit
SRMODE      = SR<13>,          ! Mode Selection Bit
SRCARRY     = SR<0>,           ! Carry Bit
SROVER      = SR<1>,           ! Overflow Bit
SRZERO      = SR<2>,           ! Zero Bit
SRNEG       = SR<3>,           ! Negative Bit
SREX        = SR<4>,           ! Extend Bit
SRMASK      = SR<10:8>,        ! Interrupt Mask
FCSPACE     = FC<1:0>,         ! Memory Access Address Space
FCMODE      = FC<2>,           ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,        ! PC Low Word
PCHI        = PC<31:16>,       ! PC High Word
D0LWORD     = D[0]<15:0>,      ! D[0] Low Word
D1LWORD     = D[1]<15:0>,      ! D[1] Low Word
D2LWORD     = D[2]<15:0>,      ! D[2] Low Word
D3LWORD     = D[3]<15:0>,      ! D[3] Low Word
D4LWORD     = D[4]<15:0>,      ! D[4] Low Word
D5LWORD     = D[5]<15:0>,      ! D[5] Low Word
D6LWORD     = D[6]<15:0>,      ! D[6] Low Word
D7LWORD     = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                    16K 16-Bit Word Internal Memory                 */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                   Logic Level Macros                               */
/*                                                                   */
/*********************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*********************************************************************/
/*                                                                   */
/*  Power On and Initialization. This process was not modeled but is  */
/*  added to initialize signals and registers.                       */
/*                                                                   */
/*********************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                ! Place Memory Locations Following The
        M[0x100f] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready.
        /***********************************************************/
        /*                                                         */
        /*      Routine Initialization Per Hamby and Guillory      */
        /*                                                         */
        /***********************************************************/
        D[1] = 0x5555;              ! Place Hex 5555 Into D[1]
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/*********************************************************************/
/*                                                                   */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                       */
/*                                                                       */
/***********************************************************************/

fetch_initial_instruction :=
      (

      /*****************************************************************/

      PHI1 = hi;                                    ! Phase 1 Of
      PHI2 = lo;                                    ! Clock Cycle 0
      RW = hi;                                      ! Memory Read
      ADENABLE = lo;                                ! Disable Address Bus Buffer
      DBENABLE = lo;                                ! Disable Data Bus Buffer
      IABUS = PC;                                   ! Place PC On Internal Address
                                                    ! Bus
      next;                                         ! Execute Pending Assignments

      PHI1 = lo;                                    ! Phase 2 Of
      PHI2 = hi;                                    ! Clock Cycle 0
      ADENABLE = hi;                                ! Enable Address Bus Buffer
      EXABUF = IABUS;                               ! Gate Internal Address Bus
                                                    ! Into External Address Buffer
      FCMODE = SRMODE;                              ! User Mode
      FCSPACE = 2;                                  ! Accessing Program
      next;                                         ! Execute Impending Assignments
      ABUS = EXABUF;                                ! Address Placed On Bus(Added)
      next;                                         ! Execute Pending Assignments

      /*****************************************************************/
      T = 1;                                        ! Clock Cycle 1
      next;                                         ! Execute Assignment

      PHI1 = hi;                                    ! Phase 1 Of
      PHI2 = lo;                                    ! Clock Cycle 1
      ASN = lo;                                     ! Assert Address Strobe
      LDSN = lo;                                    ! Assert Lower Data Strobe
      UDSN = lo;                                    ! Assert Upper Data Strobe
      DBENABLE = hi;                                ! Enable Data Bus
      next;                                         ! Execute Pending Assignments

      PHI1 = lo;                                    ! Phase 2
      PHI2 = hi;                                    ! Of Clock Cycle 1
      next;                                         ! Execute Pending Assignments

      /*****************************************************************/
      T = 2;                                        ! Clock Cycle 2
      next;                                         ! Execute Assignment

      PHI1 = hi;                                    ! Phase 1
      PHI2 = lo;                                    ! Of Clock Cycle 2
      while DTACKN eql hi                           ! Wait For Memory To Place
            (                                       ! Data On The Bus
        next;                                   ! Execute Impending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 2
        next;                                   ! Execute Assignments

        /***********************************************************/
        T = 3;                                  ! Clock Cycle 3
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
        DTACKN = lo;                            ! Asserts DTACKN(Aaded)
        next;                                   ! Execute Pending Assignments

        /***********************************************************/
        T = 2                                   ! Return To  Phase 2
                                                ! Of Clock Cycle 2
        );
        next;                                   ! Execute Impending Assignments

    /***************************************************************/
    T = 3;                                      ! Clock Cycle 3
    next;                                       ! Execute Assignment

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 3
    EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
    next;                                       ! Execute Pending Assignments

    /***************************************************************/
    T = 4;                                      ! Clock Cycle 4
    next;                                       ! Execute Assignment

    PHI1 = hi;                                  ! Phase 1
    PHI2 = lo;                                  ! Of Clock Cycle 4
    PFR = EXDBUF;                               ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
    next;                                       ! Execute Pending Assignments

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 4
    ASN = hi;                                   ! Deactivate Address Strobe
    LDSN = hi;                                  ! Deactivate Lower Data Strobe
    UDSN = hi;                                  ! Deactivate Upper Data Strobe
    IR = PFR;                                   ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                                  ! Acknowledge
        PC = PC + 4;                              ! Increment Program Counter
        next;                                     ! Execute Pending Assignments
        T = 0                                     ! Reset Clock Cycle Counter
        )

andi :=                                           ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                               ! Effect Of Instruction
       IR<15:8> = M[PC];                          ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                      ! Is To Set Status Register
          PC = PC + 2;                              ! Increment Program Counter
       T = 5;                                     ! Supervisor Bit To User
       next;                                      ! Mode
       T = 0                                      ! Requires 6 Clock Cycles
       )

move :=                                           ! MOVE.W D1,$2004 [2008]
        (

        /****************************************************************/

        PHI1 = hi;                                ! Phase 1 Of
        PHI2 = lo;                                ! Clock Cycle 0
        DBUS = 0xffff;                            ! Place Data Bus In High Impedance
        RW = hi;                                  ! Memory Read
        ADENABLE = lo;                            ! Disable Address Bus Buffer
        ABUS = 0xffffff;                          ! Address Bus High Impedanced
        DBENABLE = lo;                            ! Disable Data Bus Buffer
        IABUS<31:1> = PC<31:1>;                   ! Place PC On Internal Address
                                                  ! Bus
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2 Of
        PHI2 = hi;                                ! Clock Cycle 0
        ADENABLE = hi;                            ! Enable Address Bus Buffer
        EXABUF = IABUS<23:1>;                     ! Gate Internal Address Bus
                                                  ! Into External Address Buffer
        FCMODE = SRMODE;                          ! User Mode
        FCSPACE = 2;                              ! Accessing Program
        ABUS = IABUS<23:1>;                       ! Address Placed On Bus
        next;                                     ! Execute Impending Assignments

        /****************************************************************/
        T = 1;                                    ! Clock Cycle 1
        next;                                     ! Execute Assignment

        PHI1 = hi;                                ! Phase 1 Of
        PHI2 = lo;                                ! Clock Cycle 1
        ASN = lo;                                 ! Assert Address Strobe
        LDSN = lo;                                ! Assert Lower Data Strobe
```

```
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                        ! Phase 1
PHI2 = lo;                        ! Of Clock Cycle 4
DISREG = EXDBUF sxt 32;           ! Store Displacement
next;                             ! Execute Pending Assignments

PHI1 = lo;                        ! Phase 2
PHI2 = hi;                        ! Of Clock Cycle 4
ASN = hi;                         ! Deactivate Address Strobe
LDSN = hi;                        ! Deactivate Lower Data Strobe
UDSN = hi;                        ! Deactivate Upper Data Strobe.
                                  ! Are Placed Into Instruction
                                  ! Register
PC = PC + 2;                      ! Increment Program Counter
DTACKN = hi;                      ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
next;

/**************************************************************/
T = 5;                            ! Clock Cycle 5
next;                             ! Execute Previous Assignment

PHI1 = hi;                        ! Phase 1 Of
PHI2 = lo;                        ! Clock Cycle 5
RW = hi;                          ! Memory Read
ADENABLE = lo;                    ! Disable Address Bus Buffer
ABUS = 0xffffff;                  ! Address Bus High Impedanced
DBUS = 0xffff;                    ! Data Bus Returned To High
                                  ! Impedance State
DBENABLE = lo;                    ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;           ! Place PC On Internal Address
                                  ! Bus
next;                               Execute Pending Assignments

PHI1 = lo;                        ! Phase 2 Of
PHI2 = hi;                        ! Clock Cycle 5
ADENABLE = hi;                    ! Enable Address Bus Buffer
FCMODE = SRMODE;                  ! User Mode
FCSPACE = 2;                      ! Accessing Data
EXABUF = IABUS<23:1>;             ! Gate Internal Address Bus
ABUS = IABUS<23:1>;               ! Place Address On Bus
next;                             ! Into External Address Buffer

/**************************************************************/
T = 6;                            ! Clock Cycle 6
next;                             ! Execute Assignment

PHI1 = hi;                        ! Phase 1 Of
PHI2 = lo;                        ! Clock Cycle 6
UDSN = lo;                        ! Activate Upper And
LDSN = lo;                        ! Lower Data Strobes
ASN = lo;                         ! Assert Address Strobe
DBENABLE = hi;                    ! Enable Data Bus
```

```
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 6
next;                                 ! Execute Pending Assignments

/*******************************************************************/
T = 7;                                ! Clock Cycle 7
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 7
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 7
      next;                           ! Execute Assignments

      /*************************************************************/
      T = 8;                          ! Clock Cycle 8
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 8
      DBUS<15:8> = MEABUS];           ! Memory Places Instruction
      DBUS<7:0> = MEABUS + 1];        ! On Data Bus And
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /*************************************************************/
      T = 7                           ! Return To  Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments
/*******************************************************************/
T = 8;                                ! Clock Cycle 8
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 8
EXDBUF = DBUS;                        ! Instruction On Data Bus
                                      ! Is Placed In External Data
                                      ! Bus Buffer
next;                                 ! Execute Pending Assignments

/*******************************************************************/
T = 9;                                ! Clock Cycle 9
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
```

```
PHI2 = lo;                              ! Of Clock Cycle 9
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 9
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer(Aaded)
                                        ! Acknowledge
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 10;                                  ! Clock Cycle 10
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 10
DBUS = 0xffff;                          ! Place Data Bus In High Impedance
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = DISREG;                         ! Place DISREG On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 10
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                        ! Into External Address Buffer
IDBUS = D1LWORD;                        ! Place Low Word Of D[1] On Bus
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Data
ABUS = IABUS<23:1>;                     ! Address Placed On Bus
next;                                   ! Execute Impending Assignments

/*****************************************************************/
T = 11;                                  ! Clock Cycle 11
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 11
ASN = lo;                               ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                         ! Place Contents Of Internal
                                        ! Data Bus Into External Data Buffer
SRCARRY = lo;                           ! Reset Condition Code Bits
SROVER = lo;
```

```
SRZERO = lo;
SRNEG = lo;
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 11
if EXDBUF eql 0                             ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                              ! Place Data On External Data Bus
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 12;                                      ! Clock Cycle 12
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 12
if EXDBUF<15>                               ! Set Negative Condition Bit
   SRNEG = hi;                              ! If Needed
UDSN = lo;                                  ! Activate Upper And
LDSN = lo;                                  ! Lower Data Strobes
twait = 0;                                  ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      twait = twait + 1;                    ! Increment Wait Cycle
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 12
      next;                                 ! Execute Assignments

      /************************************************************/
      T = 13;                                ! Clock Cycle 13
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 13
      if twait eql 2                        ! Memory Responds After 2 Cycles
      (
      M[ABUS] = DBUS<15:8>;                 ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;              ! In Memory
      DTACKN = lo                         ! Asserts DTACKN(Added)
      );
      next;                                 ! Execute Pending Assignments

      /************************************************************/
      T = 12                                ! Return To Phase 2
                                          ! Of Clock Cycle 12
      );
      next;                               ! Execute Impending Assignments
```

```
        /*****************************************************************/
        T = 13;                                  ! Clock Cycle 13
        next;                                    ! Execute Assignment

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 13
        next;                                    ! Execute Pending Assignments

        /*****************************************************************/
        T = 14;                                  ! Clock Cycle 14
        next;                                    ! Execute Assignment

        PHI1 = hi;                               ! Phase 1
        PHI2 = lo;                               ! Of Clock Cycle 14
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 9
        ASN = hi;                                ! Deactivate Address Strobe
        LDSN = hi;                               ! Deactivate Lower Data Strobe
        UDSN = hi;                               ! Deactivate Upper Data Strobe
        PC = PC + 2;                             ! Increment Program Counter
        IR = PFR;                                ! Place Contents Of Prefetch
                                                 ! Register Into Instruction
                                                 ! Register
        DTACKN = hi;                             ! Deactivate Data Transfer
                                                 ! Acknowledge(Added)
        next;                                    ! Execute Pending Assignments
        T = 0
        )

    jmp :=                                       ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                               ! Phase 1 Of
        PHI2 = lo;                               ! Clock Cycle 0
        DBUS = 0xffff;                           ! Place Data Bus In A High Impedance
        RW = hi;                                 ! Memory Read
        ADENABLE = lo;                           ! Disable Address Bus Buffer
        DBENABLE = lo;                           ! Disable Data Bus Buffer
        IABUS = PC;                              ! Place PC On Internal Address
                                                 ! Bus
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2 Of
        PHI2 = hi;                               ! Clock Cycle 0
        ADENABLE = hi;                           ! Enable Address Bus Buffer
        EXABUF = IABUS;                          ! Gate Internal Address Bus
                                                 ! Into External Address Buffer
        FCMODE = SRMODE;                         ! User Mode
        FCSPACE = 2;                             ! Accessing Program
```

```
next;                                   ! Execute Pending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*************************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/*************************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*******************************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*******************************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
```

```
     next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                 ! Clock Cycle 3
next;                                  ! Execute Assignment

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 3
EXDBUF = DBUS;                         ! Instruction On Data Bus
                                       ! Is Placed In External Data
                                       ! Bus Buffer
next;                                  ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                 ! Clock Cycle 4
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 4
next;
PFR = EXDBUF;                          ! The Contents Of The External
                                       ! Data Bus Buffer Are Placed
                                       ! In Prefetch Register
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 4
ASN = hi;                              ! Deactivate Address Strobe
LDSN = hi;                             ! Deactivate Lower Data Strobe
UDSN = hi;                             ! Deactivate Upper Data Strobe
DTACKN = hi;                           ! Deactivate Data Transfer
                                       ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                 ! Clock Cycle 5
next;                                  ! Execute Previous Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 5
RW = hi;                               ! Memory Read
ADENABLE = lo;                         ! Disable Address Bus Buffer
DBENABLE = lo;                         ! Disable Data Bus Buffer
IABUS = PC;                            ! Place PC On Internal Address
                                       ! Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2 Of
PHI2 = hi;                             ! Clock Cycle 5
ADENABLE = hi;                         ! Enable Address Bus Buffer
FCMODE = SRMODE;                       ! User Mode
FCSPACE = 2;                           ! Accessing Program
EXABUF = IABUS;                        ! Gate Internal Address Bus
next;                                  ! Into External Address Buffer
```

```
ABUS = EXABUF;                              ! Address Placed On Bus(Added)
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                      ! Clock Cycle 6
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 6
ASN = lo;                                   ! Assert Address Strobe
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /*********************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /*********************************************************/
      T = 7                                ! Return To Phase 2
                                           ! Of Clock Cycle 7
      );
      next;                                ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment
```

```
        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 8
        EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                    ! Is Placed In External Data
                                                    ! Bus Buffer
        next;                                       ! Execute Pending Assignments

        /****************************************************************/
        T = 9;                                      ! Clock Cycle 9
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1
        PHI2 = lo;                                  ! Of Clock Cycle 9
        PFR = EXDBUF;                               ! The Contents Of The External
                                                    ! Data Bus Buffer Are Placed
                                                    ! In Prefetch Register
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 9
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        PC = PC + 2;                                ! Increment Program Counter
        IR = PFR;                                   ! Place Contents Of Prefetch
                                                    ! Register Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer
                                                    ! Acknowledge(Added)
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                            (
                            case IR
                                 0x31c1: move       ! MOVE.W D1,$2004 [2008]
                                 0x027c: andi       ! AND.W #$DFFF,SR
                                 047320: jmp        ! JMP (A0) If IR = Octal Value
                            esac
                            )

    main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
              (
              decode_execute_prefetch
              )
        )
```

```
/************************************************************************/
/*                                                                      */
/*    MOTOROLA MC68000 MODEL OF THE MOVE.W A1,D3 INSTRUCTION            */
/*                                                                      */
/************************************************************************/

/************************************************************************/
/*                                                                      */
/*                   Structure Declarations                             */
/*                                                                      */
/************************************************************************/

state

/************************************************************************/
/*                                                                      */
/*               M68000 Programming Registers                           */
/*                                                                      */
/************************************************************************/

D[0:7]<31:0>,                ! 8 Data Registers
A[0:6]<31:0>,                ! 7 Address Registers
UA7<31:0>,                   ! User Stack Pointer
SA7<31:0>,                   ! System Stack Pointer
PC<31:0>,                    ! Program Counter
SR<15:0>,                    ! Status Register

/************************************************************************/
/*                                                                      */
/*               Temporary Internal Registers                           */
/*                                                                      */
/************************************************************************/

PFR<15:0>,                   ! Prefetch Register
IR<15:0>,                    ! Instruction Register
FC<2:0>,                     ! Function Code Register
EXDBUF<15:0>,                ! External Data Bus Buffer Register
EXABUF<23:1>,                ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,               ! ALU Buffer 1
ALUBUF2<31:0>,               ! ALU Buffer 2
DTEMP<15:0>,                 ! Temporary Data Storage
DISREG<31:0>,                ! Temporary Displacement Storage
SRTEMP<15:0>,                ! Temporary Status Register Storage
                             ! (Exception Processing)
IRTEMP<15:0>,                ! Temporary Instruction Register Storage
                             ! (Exception Processing)
TEMPADR<31:0>,               ! Temporary Cycle Address Storage
                             ! (Exception Processing)
ACTYPE<15:0>,                ! Temporary Access Type Storage
                             ! (Exception Processing)
VECADR<23:0>,                ! Temporary Vector Address Storage
                             ! (Exception Processing)
```

```
HANADR<31:0>,                  ! Temporary Address Storage For
                               ! Exception Handler Routine
T<7:0>,                        ! Clock Cycle Counter
RESET,                       ! Reset Flip-Flop
HALT,                        ! Halt Flip-Flop
RW,                          ! Read/Write Flip-Flop
ADENABLE,                    ! Address Bus Buffer Enable
DBENABLE,                    ! Data Bus Buffer Enable
ASN,                         ! Address Strobe Flip-Flop
LDSN,                        ! Lower Data Strobe Flip-Flop
UDSN,                        ! Upper Data Strobe Flip-Flop
DTACKN,                      ! Data Transfer Acknowledge Flip-Flop
COUT,                        ! Carry Flip-Flop
EXCEPT,                      ! Exception Processing Flip-Flop
READY,                       ! Ready Flip-Flop

/*********************************************************************/
/*                                                                   */
/*      Model transformation modifications:                          */
/*                                                                   */
/*           1) CDL decoder structure nonexistent in ISP' and un-    */
/*      necessary for model. Eliminated.                             */
/*           2) Multi-phase clock structure nonexistent in ISP'.     */
/*      Operations on registers will provide its equivalent.         */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                        */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                  */
/*           5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing   */
/*      MC68000 assembler and linker/loader models. The memory was   */
/*      also reduced from 8 Mwords to 32 Kbytes.                     */
/*                                                                   */
/*********************************************************************/

IABUS<31:0>,                   ! Internal Address Bus
IDBUS<31:0>,                   ! Internal Data Bus
SWITCH,                      ! Power Switch
PHI1,                        ! Phase 1 Of Two-Phase Clock
PHI2;                        ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                   */
/*                External Address and Data Bus                      */
/*                                                                   */
/*********************************************************************/

DBUS<15:0>,                    ! External Data Bus
ABUS<23:1>;                    ! External Address Bus(changed)
```

```
format

/*************************************************************************/
/*                                                                       */
/*                       Register Subfields                              */
/*                                                                       */
/*************************************************************************/

PCADDR       = PC<23:0>,        ! Program Counter Address Field
SRTRACE      = SR<15>,          ! Trace Bit
SRMODE       = SR<13>,          ! Mode Selection Bit
SRCARRY      = SR<0>,           ! Carry Bit
SROVER       = SR<1>,           ! Overflow Bit
SRZERO       = SR<2>,           ! Zero Bit
SRNEG        = SR<3>,           ! Negative Bit
SREX         = SR<4>,           ! Extend Bit
SRMASK       = SR<10:8>,        ! Interrupt Mask
FCSPACE      = FC<1:0>,         ! Memory Access Address Space
FCMODE       = FC<2>,           ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,        ! PC Low Word
PCHI         = PC<31:16>,       ! PC High Word
A1LWORD      = A[1]<15:0>,      ! A[1] Low Word
D0LWORD      = D[0]<15:0>,      ! D[0] Low Word
D1LWORD      = D[1]<15:0>,      ! D[1] Low Word
D2LWORD      = D[2]<15:0>,      ! D[2] Low Word
D3LWORD      = D[3]<15:0>,      ! D[3] Low Word
D4LWORD      = D[4]<15:0>,      ! D[4] Low Word
D5LWORD      = D[5]<15:0>,      ! D[5] Low Word
D6LWORD      = D[6]<15:0>,      ! D[6] Low Word
D7LWORD      = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/*************************************************************************/
/*                                                                       */
/*                   16K 16-Bit Word Internal Memory                     */
/*                                                                       */
/*************************************************************************/

M[0:32767]<7:0>;

macro

/*************************************************************************/
/*                                                                       */
/*                       Logic Level Macros                              */
/*                                                                     */
/***********************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/***********************************************************************/
/*                                                                     */
/*  Power On and Initialization. This process was not modeled but is  */
/*  added to initialize signals and registers.                         */
/*                                                                     */
/***********************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                     ! Turn Power On
        next;                            ! Execute Assignment
        READY = lo;                      ! System Not Ready
        RESET = lo;                      ! Assert Reset For
        delay(100);                      ! 100 Miliseconds(Active Low)
        RESET = hi;                      ! Deactivate Reset
        next;                            ! Execute Pending Assignments
        ASN = hi;                        ! Initialize Address Strobe
        LDSN = hi;                       ! Initialize Lower Data Strobe
        UDSN = hi;                       ! Initialize Upper Data Strobe
        DTACKN = hi;                     ! Initialize Data Transfer Acknowledge
        RW = hi;                         ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[0x100a] = 0xff;                 ! Place Memory Locations Following The
        M[0x100b] = 0xff;                  ! JMP Instruction In A High State
        HALT = hi;                       ! Initialize Halt Flip-Flop(Active
                                         ! Low)
        T = 0;                           ! Initialize Clock Cycle Counter
        READY = hi;                      ! System Ready
        /***********************************************************/
        /*                                                         */
        /*     Routine Initialization Per Hamby and Guillory       */
        /*                                                         */
        /***********************************************************/
        A[1] = 0x55555555;               ! Place Hex 55555555 Into A[1]
        A[0] = 0x1004;                   ! Place Hex 1004 Into A[0]
        PC = 0x1000;                     ! Place Hex 1000 Into Program Counter
        next                             ! Execute Assignments
        )

/***********************************************************************/
/*                                                                     */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /*****************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /*****************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /*****************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
      while DTACKN eql hi                     ! Wait For Memory To Place
            (                                 ! Data On The Bus
          next;                          ! Execute Impending Assignments

          PHI1 = lo;                     ! Phase 2
          PHI2 = hi;                     ! Of Clock Cycle 2
          next;                          ! Execute Assignments

          /*****************************************************************/
          T = 3;                         ! Clock Cycle 3
          next;                          ! Execute Assignment

          PHI1 = hi;                     ! Phase 1
          PHI2 = lo;                     ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];          ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];       ! On Data Bus And
          DTACKN = lo;                   ! Asserts DTACKN(Added)
          next;                          ! Execute Pending Assignments

          /*****************************************************************/
          T = 2                          ! Return To  Phase 2
                                         ! Of Clock Cycle 2
          );
          next;                          ! Execute Impending Assignments

      /***********************************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 3
      EXDBUF = DBUS;                     ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
      next;                              ! Execute Pending Assignments

      /***********************************************************************/
      T = 4;                             ! Clock Cycle 4
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 4
      PFR = EXDBUF;                      ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
      next;                              ! Execute Pending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 4
      ASN = hi;                          ! Deactivate Address Strobe
      LDSN = hi;                         ! Deactivate Lower Data Strobe
      UDSN = hi;                         ! Deactivate Upper Data Strobe
      IR = PFR;                          ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
```

```
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 4;                            ! Increment Program Counter
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )



move  :=                                        ! MOVE.W A1,D3
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        IDBUS = A1LWORD;                        ! Place Low Word From A[1] Onto
                                                ! Internal Data Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        SRCARRY = lo;                           ! Clear Status Register Carry Bit
        SROVER = lo;                            ! Clear Status Register Overflow Bit
        SRZERO = lo;                            ! Clear Status Register Zero Bit
        SRNEG  = lo;                            ! Clear Status Register Negative Bit
        ABUS = IABUS;                           ! Place PC On Address Bus (Added)
        D3LWORD = IDBUS;                        ! Place Data From Internal Data Bus
                                                ! Into Low Word Of D[3]
        next;                                   ! Execute Impending Assignments

        /*****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        DBENABLE = hi;                          ! Enable Data Bus
        if D3LWORD eql 0                        ! Set Status Register Zero Bit
```

```
   SRZERO = hi;                         ! If Moved Data Is Zero
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
if D[3]<15>                             ! Set Status Register Negative
   SRNEG = hi;                          ! Bit If Moved Data Is Negative
next;                                   ! Execute Pending Assignments

/*********************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***************************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***************************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*********************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*********************************************************************/
T = 4;                                  ! Clock Cycle 4
```

1/6

```
        next;                                    ! Execute Assignment

        PHI1 = hi;                               ! Phase 1
        PHI2 = lo;                               ! Of Clock Cycle 4
        PFR = EXDBUF;                            ! The Contents Of The External
                                                 ! Data Bus Buffer Are Placed
                                                 ! In Prefetch Register
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 4
        ASN = hi;                                ! Deactivate Address Strobe
        LDSN = hi;                               ! Deactivate Lower Data Strobe
        UDSN = hi;                               ! Deactivate Upper Data Strobe
        IR = PFR;                                ! Contents Of Prefetch Register
                                                 ! Are Placed Into Instruction
                                                 ! Register
        DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                                 ! Acknowledge
        PC = PC + 2;                             ! Increment Program Counter
        next;                                    ! Execute Impending Assignments
        T = 0                                    ! Reset Clock Cycle Counter
        )

jmp :=                                           ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                               ! Phase 1 Of
        PHI2 = lo;                               ! Clock Cycle 0
        DBUS = 0xffff;                           ! Place Data Bus In A High Impedance
        RW = hi;                                 ! Memory Read
        ADENABLE = lo;                           ! Disable Address Bus Buffer
        DBENABLE = lo;                           ! Disable Data Bus Buffer
        IABUS = PC;                              ! Place PC On Internal Address
                                                 ! Bus
        next;                                    ! Execute Pending Assignments

        PHI1 = lo;                               ! Phase 2 Of
        PHI2 = hi;                               ! Clock Cycle 0
        ADENABLE = hi;                           ! Enable Address Bus Buffer
        EXABUF = IABUS;                          ! Gate Internal Address Bus
                                                 ! Into External Address Buffer
        FCMODE = SRMODE;                         ! User Mode
        FCSPACE = 2;                             ! Accessing Program
        next;                                    ! Execute Pending Assignments
        ABUS = EXABUF;                           ! Address Placed On Bus(Added)
        next;                                    ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                   ! Clock Cycle 1
        next;                                    ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /**********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /**********************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
```

```
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                           ! Phase 1 Of
PHI2 = lo;                           ! Clock Cycle 6
ASN = lo;                            ! Assert Address Strobe
LDSN = lo;                           ! Assert Lower Data Strobe
UDSN = lo;                           ! Assert Upper Data Strobe
DBENABLE = hi;                       ! Enable Data Bus
next;                                ! Execute Pending Assignments

PHI1 = lo;                           ! Phase 2
PHI2 = hi;                           ! Of Clock Cycle 6
next;                                ! Execute Pending Assignments

/*************************************************************/
T = 7;                               ! Clock Cycle 7
next;                                ! Execute Assignment

PHI1 = hi;                           ! Phase 1
PHI2 = lo;                           ! Of Clock Cycle 7
while DTACKN eql hi                  ! Wait For Memory To Place
    (                                ! Data On The Bus
    next;                            ! Execute Impending Assignments

    PHI1 = lo;                       ! Phase 2
    PHI2 = hi;                       ! Of Clock Cycle 7
    next;                            ! Execute Assignments

    /*********************************************************/
    T = 8;                           ! Clock Cycle 8
    next;                            ! Execute Assignment

    PHI1 = hi;                       ! Phase 1
    PHI2 = lo;                       ! Of Clock Cycle 8
    DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
    DTACKN = lo;                     ! Asserts DTACKN(Added)
    next;                            ! Execute Pending Assignments

    /*********************************************************/
    T = 7                           ! Return To Phase 2
                                    ! Of Clock Cycle 7
    );
    next;                           ! Execute Impending Assignments

/*************************************************************/
T = 8;                               ! Clock Cycle 8
next;                                ! Execute Assignment

PHI1 = lo;                           ! Phase 2
PHI2 = hi;                           ! Of Clock Cycle 8
EXDBUF = DBUS;                       ! Instruction On Data Bus
                                     ! Is Placed In External Data
                                     ! Bus Buffer
next;                                ! Execute Pending Assignments
```

```
        /*****************************************************************/
        T = 9;                                    ! Clock Cycle 9
        next;                                     ! Execute Assignment

        PHI1 = hi;                                ! Phase 1
        PHI2 = lo;                                ! Of Clock Cycle 9
        PFR = EXDBUF;                             ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Prefetch Register
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 9
        ASN = hi;                                 ! Deactivate Address Strobe
        LDSN = hi;                                ! Deactivate Lower Data Strobe
        UDSN = hi;                                ! Deactivate Upper Data Strobe
        PC = PC + 2;                              ! Increment Program Counter
        IR = PFR;                                 ! Place Contents Of Prefetch
                                                  ! Register Into Instruction
                                                  ! Register
        DTACKN = hi;                              ! Deactivate Data Transfer
                                                  ! AcKnowledge(Added)
        next;                                     ! Execute Pending Assignments
        T = 0                                     ! Reset Clock Cycle Counter
        )

    andi :=                                       ! AND.W ##DFFF,SR
       (
       SRMODE = lo;                               ! Set Status Register To
       IR<15:8> = M[PC];                          ! User Mode And Prefetch
       IR<7:0> = M[PC + 1];                       ! Next Instruction
       next;
       PC = PC + 2;                               ! Increment PC
       T = 5;                                     ! Requires 6 Clock Cycles
       next;
       T = 0
       )

    decode_execute_prefetch :=
                          (
                          case IR
                               0x3609: move       ! MOVE.W A1,D3
                               0x027c: andi       ! AND.W #DFFF,SR
                               047320: jmp        ! JMP (A0) If IR = Octal Value
                          esac
                          )

    main :=
       (
       power_on_initialize;
       fetch_initial_instruction;
       while READY eql hi
```

```
                (
                decode_execute_prefetch
                )
        )
```

```
/***********************************************************************/
/*                                                                     */
/*    MOTOROLA MC68000 MODEL OF THE MOVE.W (A1),D2 INSTRUCTION        */
/*                                                                     */
/***********************************************************************/

/***********************************************************************/
/*                                                                     */
/*                    Structure Declarations                          */
/*                                                                     */
/***********************************************************************/

state

/***********************************************************************/
/*                                                                     */
/*                 M68000 Programming Registers                       */
/*                                                                     */
/***********************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/***********************************************************************/
/*                                                                     */
/*                 Temporary Internal Registers                       */
/*                                                                     */
/***********************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,               ! Temporary Address Storage For
                            ! Exception Handler Routine
T<7:0>,                     ! Clock Cycle Counter
RESET,                   ! Reset Flip-Flop
HALT,                    ! Halt Flip-Flop
RW,                      ! Read/Write Flip-Flop
ADENABLE,                ! Address Bus Buffer Enable
DBENABLE,                ! Data Bus Buffer Enable
ASN,                     ! Address Strobe Flip-Flop
LDSN,                    ! Lower Data Strobe Flip-Flop
UDSN,                    ! Upper Data Strobe Flip-Flop
DTACKN,                  ! Data Transfer Acknowledge Flip-Flop
COUT,                    ! Carry Flip-Flop
EXCEPT,                  ! Exception Processing Flip-Flop
READY,                   ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                           */
/*                                                                    */
/*           1) CDL decoder structure nonexistent in ISP' and un-     */
/*      necessary for model. Eliminated.                              */
/*           2) Multi-phase clock structure nonexistent in ISP'.      */
/*      Operations on registers will provide its equivalent.          */
/*           3) Switch structure nonexistent in ISP'. Operation on a  */
/*      register will provide its equivalent.                         */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*           5) The memory word length was reduced from 16 to 8 bit   */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,                ! Internal Address Bus
IDBUS<31:0>,                ! Internal Data Bus
twait<7:0>,                 ! Wait Cycle Counter
SWITCH,                  ! Power Switch
PHI1,                    ! Phase 1 Of Two-Phase Clock
PHI2;                    ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                 ! External Data Bus
```

```
ABUS<23:1>;                    ! External Address Bus(changed)

format

/************************************************************************/
/*                                                                      */
/*                      Register Subfields                             */
/*                                                                      */
/************************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/************************************************************************/
/*                                                                      */
/*                      16K 16-Bit Word Internal Memory                */
/*                                                                      */
/************************************************************************/

M[0:32767]<7:0>;

macro

/************************************************************************/
/*                                                                      */
/*                     Logic Level Macros                              */
/*                                                                    */
/**********************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/**********************************************************************/
/*                                                                    */
/*  Power On and Initialization. This process was not modeled but is  */
/*  added to initialize signals and registers.                        */
/*                                                                    */
/**********************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                     ! Turn Power On
        next;                            ! Execute Assignment
        READY = lo;                      ! System Not Ready
        RESET = lo;                      ! Assert Reset For
        delay(100);                      ! 100 Miliseconds(Active Low)
        RESET = hi;                      ! Deactivate Reset
        next;                            ! Execute Pending Assignments
        ASN = hi;                        ! Initialize Address Strobe
        LDSN = hi;                       ! Initialize Lower Data Strobe
        UDSN = hi;                       ! Initialize Upper Data Strobe
        DTACKN = hi;                     ! Initialize Data Transfer Acknowledge
        RW = hi;                         ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[0x100a] = 0xff;                ! Place Memory Locations Following The
        M[0x100b] = 0xff;                  ! JMP Instruction In A High State
        HALT = hi;                       ! Initialize Halt Flip-Flop(Active
                                         ! Low)
        T = 0;                           ! Initialize Clock Cycle Counter
        READY = hi;                      ! System Ready
        /**************************************************************/
        /*                                                            */
        /*      Routine Initialization Per Hamby and Guillory         */
        /*                                                            */
        /**************************************************************/
        M[0x2000] = 0x55;                ! Initialize Memory Location
        M[0x2001] = 0x55;                ! 2000 Hex To 5555 Hex
        A[0] = 0x1004;                   ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;                   ! Store Data At Hex 2000
        PC = 0x1000;                     ! Place Hex 1000 Into Program Counter
        next                             ! Execute Assignments
        )

/**********************************************************************/
/*                                                                    */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary  */
/*  to retrieve modeled instructions for simulation and analysis. It    */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /*****************************************************************/

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 0
      RW = hi;                            ! Memory Read
      ADENABLE = lo;                      ! Disable Address Bus Buffer
      DBENABLE = lo;                      ! Disable Data Bus Buffer
      IABUS = PC;                         ! Place PC On Internal Address
                                          ! Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2 Of
      PHI2 = hi;                          ! Clock Cycle 0
      ADENABLE = hi;                      ! Enable Address Bus Buffer
      EXABUF = IABUS;                     ! Gate Internal Address Bus
                                          ! Into External Address Buffer
      FCMODE = SRMODE;                    ! User Mode
      FCSPACE = 2;                        ! Accessing Program
      next;                               ! Execute Impending Assignments
      ABUS = EXABUF;                      ! Address Placed On Bus(Added)
      next;                               ! Execute Pending Assignments

      /*****************************************************************/
      T = 1;                              ! Clock Cycle 1
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 1
      ASN = lo;                           ! Assert Address Strobe
      LDSN = lo;                          ! Assert Lower Data Strobe
      UDSN = lo;                          ! Assert Upper Data Strobe
      DBENABLE = hi;                      ! Enable Data Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 1
      next;                               ! Execute Pending Assignments

      /*****************************************************************/
      T = 2;                              ! Clock Cycle 2
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 2
```

```
while DTACKN eql hi                         ! Wait For Memory To Place
     (                                      ! Data On The Bus
     next;                                  ! Execute Impending Assignments

     PHI1 = lo;                             ! Phase 2
     PHI2 = hi;                             ! Of Clock Cycle 2
     next;                                  ! Execute Assignments

     /*******************************************************/
     T = 3;                                 ! Clock Cycle 3
     next;                                  ! Execute Assignment

     PHI1 = hi;                             ! Phase 1
     PHI2 = lo;                             ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
     DTACKN = lo;                           ! Asserts DTACKN(Added)
     next;                                  ! Execute Pending Assignments

     /*******************************************************/
     T = 2                                  ! Return To  Phase 2
                                            ! Of Clock Cycle 2
     );
     next;                                  ! Execute Impending Assignments

/**********************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/**********************************************************/
T = 4;                                      ! Clock Cycle 4
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 4
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 4
ASN = hi;                                   ! Deactivate Address Strobe
LDSN = hi;                                  ! Deactivate Lower Data Strobe
UDSN = hi;                                  ! Deactivate Upper Data Strobe
IR = PFR;                                   ! Contents Of Prefetch Register
```

```
                                        ! Are Placed Into Instruction
                                        ! Register
    DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
    PC = PC + 4;                        ! Increment Program Counter
    next;                               ! Execute Pending Assignments
    T = 0                               ! Reset Clock Cycle Counter
    )

andi :=                                 ! AND.W #$DFFF,SR
   (
   SRMODE = lo;                         ! Effect Of Instruction
   IR<15:8> = M[PC];                    ! Prefetch Next Instruction
   IR<7:0> = M[PC + 1];
   next;                                ! Is To Set Status Register
      PC = PC + 2;                        ! Increment Program Counter
   T = 5;                               ! Supervisor Bit To User
   next;                                ! Mode
   T = 0                                ! Requires 6 Clock Cycles
   )

move :=                                 ! MOVE.W (A1),D2
    (

    /*****************************************************************/

    PHI1 = hi;                          ! Phase 1 Of
    PHI2 = lo;                          ! Clock Cycle 0
    DBUS = 0xffff;                      ! Place Data Bus In High Impedance
    RW = hi;                            ! Memory Read
    ADENABLE = lo;                      ! Disable Address Bus Buffer
    DBENABLE = lo;                      ! Disable Data Bus Buffer
    IABUS = PC;                         ! Place PC On Internal Address
                                        ! Bus
    next;                               ! Execute Pending Assignments

    PHI1 = lo;                          ! Phase 2 Of
    PHI2 = hi;                          ! Clock Cycle 0
    ADENABLE = hi;                      ! Enable Address Bus Buffer
    EXABUF = IABUS;                     ! Gate Internal Address Bus
                                        ! Into External Address Buffer
    FCMODE = SRMODE;                    ! User Mode
    FCSPACE = 2;                        ! Accessing Program
    ABUS = IABUS;                       ! Place PC On Address Bus
    next;                               ! Execute Impending Assignments

    /*****************************************************************/
    T = 1;                              ! Clock Cycle 1
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1 Of
    PHI2 = lo;                          ! Clock Cycle 1
    ASN = lo;                           ! Assert Address Strobe
```

```
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 1
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 2;                                      ! Clock Cycle 2
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 2
while DTACKN eql hi                         ! Wait For Memory To Place
    (                                       ! Data On The Bus
    next;                                   ! Execute Impending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 2
    next;                                   ! Execute Assignments

    /*************************************************************/
    T = 3;                                  ! Clock Cycle 3
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 3
    DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
    DTACKN = lo;                            ! Asserts DTACKN(Added)
    next;                                   ! Execute Pending Assignments

    /*************************************************************/
    T = 2                                   ! Return To  Phase 2
                                            ! Of Clock Cycle 2
    );
    next;                                   ! Execute Impending Assignments

/******************************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 4;                                      ! Clock Cycle 4
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
PC = PC + 2;                            ! Increment Program Counter
                                        ! Are Placed Into Instruction
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
DBUS = 0xffff;                          ! Place Data Bus In High Impedance
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Data
SRCARRY = lo;                           ! Clear Status Register Carry Bit
SROVER = lo;                            ! Clear Status Register Overflow Bit
SRZERO = lo;                            ! Clear Status Register Zero Bit
SRNEG  = lo;                            ! Clear Status Register Negative Bit
ABUS = IABUS;                           ! Place PC On Address Bus (Added)
next;                                   ! Execute Impending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/***************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 7
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 8;                            ! Clock Cycle 8
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 7                             ! Return To  Phase 2
                                        ! Of Clock Cycle 7
      );
      next;                             ! Execute Impending Assignments

/***************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments
```

```
        /*****************************************************************/
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        IDBUS = EXDBUF;
        if EXDBUF eql 0                         ! Set Status Register
           SRZERO = hi;                         ! Bits As Appropriate
        if EXDBUF<15> eql 1
           SRNEG = hi;
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        D[2] = IDBUS;                           ! Place Contents Of Internal
                                                ! Data Bus Into D[2]
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    jmp :=                                      ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
```

```
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/***************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/***************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments
```

```
/*********************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/*********************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
next;
PFR = EXDBUF;                            ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 4
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
UDSN = hi;                               ! Deactivate Upper Data Strobe
DTACKN = hi;                             ! Deactivate Data Transfer
                                         ! Acknowledge(Added)
next;
/*********************************************************************/
T = 5;                                   ! Clock Cycle 5
next;                                    ! Execute Previous Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 5
RW = hi;                                 ! Memory Read
ADENABLE = lo;                           ! Disable Address Bus Buffer
DBENABLE = lo;                           ! Disable Data Bus Buffer
IABUS = PC;                              ! Place PC On Internal Address
                                         ! Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2 Of
PHI2 = hi;                               ! Clock Cycle 5
ADENABLE = hi;                           ! Enable Address Bus Buffer
FCMODE = SRMODE;                         ! User Mode
FCSPACE = 2;                             ! Accessing Program
EXABUF = IABUS;                          ! Gate Internal Address Bus
next;                                    ! Into External Address Buffer
ABUS = EXABUF;                           ! Address Placed On Bus(Added)
```

```
next;                                           ! Execute Pending Assignments

/*************************************************************************/
T = 6;                                          ! Clock Cycle 6
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1 Of
PHI2 = lo;                                      ! Clock Cycle 6
ASN = lo;                                       ! Assert Address Strobe
LDSN = lo;                                      ! Assert Lower Data Strobe
UDSN = lo;                                      ! Assert Upper Data Strobe
DBENABLE = hi;                                  ! Enable Data Bus
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 6
next;                                           ! Execute Pending Assignments

/*************************************************************************/
T = 7;                                          ! Clock Cycle 7
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 7
while DTACKN eql hi                             ! Wait For Memory To Place
      (                                         ! Data On The Bus
      next;                                     ! Execute Impending Assignments

      PHI1 = lo;                                ! Phase 2
      PHI2 = hi;                                ! Of Clock Cycle 7
      next;                                     ! Execute Assignments

      /*******************************************************************/
      T = 8;                                    ! Clock Cycle 8
      next;                                     ! Execute Assignment

      PHI1 = hi;                                ! Phase 1
      PHI2 = lo;                                ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                     ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                  ! On Data Bus And
      DTACKN = lo;                              ! Asserts DTACKN(Added)
      next;                                     ! Execute Pending Assignments

      /*******************************************************************/
      T = 7                                    ! Return To Phase 2
                                               ! Of Clock Cycle 7
      );
      next;                                    ! Execute Impending Assignments

/*************************************************************************/
T = 8;                                          ! Clock Cycle 8
next;                                           ! Execute Assignment
```

```
        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 8
        EXDBUF = DBUS;                          ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
        next;                                   ! Execute Pending Assignments

        /**************************************************************/
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                            (
                            case IR
                                 0x3411: move   ! MOVE.W (A1),D2
                                 0x027c: andi   ! AND.W #$DFFF,SR
                                 047320: jmp    ! JMP (A0) If IR = Octal Value
                            esac
                            )

    main :=
         (
         power_on_initialize;
         fetch_initial_instruction;
         while READY eql hi
               (
               decode_execute_prefetch
               )
         )
```

```
/**********************************************************************/
/*                                                                    */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W (A1)+,D6 INSTRUCTION      */
/*                                                                    */
/**********************************************************************/

/**********************************************************************/
/*                                                                    */
/*                     Structure Declarations                         */
/*                                                                    */
/**********************************************************************/

state

/**********************************************************************/
/*                                                                    */
/*                 M68000 Programming Registers                       */
/*                                                                    */
/**********************************************************************/

D[0:7]<31:0>,              ! 8 Data Registers
A[0:6]<31:0>,              ! 7 Address Registers
UA7<31:0>,                 ! User Stack Pointer
SA7<31:0>,                 ! System Stack Pointer
PC<31:0>,                  ! Program Counter
SR<15:0>,                  ! Status Register

/**********************************************************************/
/*                                                                    */
/*                 Temporary Internal Registers                       */
/*                                                                    */
/**********************************************************************/

PFR<15:0>,                 ! Prefetch Register
IR<15:0>,                  ! Instruction Register
FC<2:0>,                   ! Function Code Register
EXDBUF<15:0>,              ! External Data Bus Buffer Register
EXABUF<23:1>,              ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,             ! ALU Buffer 1
ALUBUF2<31:0>,             ! ALU Buffer 2
DTEMP<15:0>,               ! Temporary Data Storage
DISREG<31:0>,              ! Temporary Displacement Storage
SRTEMP<15:0>,              ! Temporary Status Register Storage
                           ! (Exception Processing)
IRTEMP<15:0>,              ! Temporary Instruction Register Storage
                           ! (Exception Processing)
TEMPADR<31:0>,             ! Temporary Cycle Address Storage
                           ! (Exception Processing)
ACTYPE<15:0>,              ! Temporary Access Type Storage
                           ! (Exception Processing)
VECADR<23:0>,              ! Temporary Vector Address Storage
                           ! (Exception Processing)
```

```
HANADR<31:0>,                  ! Temporary Address Storage For
                               ! Exception Handler Routine
T<7:0>,                        ! Clock Cycle Counter
RESET,                      ! Reset Flip-Flop
HALT,                       ! Halt Flip-Flop
RW,                         ! Read/Write Flip-Flop
ADENABLE,                   ! Address Bus Buffer Enable
DBENABLE,                   ! Data Bus Buffer Enable
ASN,                        ! Address Strobe Flip-Flop
LDSN,                       ! Lower Data Strobe Flip-Flop
UDSN,                       ! Upper Data Strobe Flip-Flop
DTACKN,                     ! Data Transfer Acknowledge Flip-Flop
COUT,                       ! Carry Flip-Flop
EXCEPT,                     ! Exception Processing Flip-Flop
READY,                      ! Ready Flip-Flop

/*******************************************************************/
/*                                                                 */
/*      Model transformation modifications:                        */
/*                                                                 */
/*           1) CDL decoder structure nonexistent in ISP' and un-  */
/*      necessary for model. Eliminated.                           */
/*           2) Multi-phase clock structure nonexistent in ISP'.   */
/*      Operations on registers will provide its equivalent.       */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                      */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                */
/*           5) The memory word length was reduced from 16 to 8 bit */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing  */
/*      MC68000 assembler and linker/loader models. The memory was  */
/*      also reduced from 8 Mwords to 32 Kbytes.                   */
/*                                                                 */
/*******************************************************************/

IABUS<31:0>,                   ! Internal Address Bus
IDBUS<31:0>,                   ! Internal Data Bus
twait<7:0>,                    ! Wait Cycle Counter
SWITCH,                     ! Power Switch
PHI1,                       ! Phase 1 Of Two-Phase Clock
PHI2;                       ! Phase 2 Of Two-Phase Clock

port

/*******************************************************************/
/*                                                                 */
/*              External Address and Data Bus                      */
/*                                                                 */
/*******************************************************************/

DBUS<15:0>,                    ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/***************************************************************************/
/*                                                                         */
/*                     Register Subfields                                  */
/*                                                                         */
/***************************************************************************/

PCADDR       = PC<23:0>,      ! Program Counter Address Field
SRTRACE      = SR<15>,        ! Trace Bit
SRMODE       = SR<13>,        ! Mode Selection Bit
SRCARRY      = SR<0>,         ! Carry Bit
SROVER       = SR<1>,         ! Overflow Bit
SRZERO       = SR<2>,         ! Zero Bit
SRNEG        = SR<3>,         ! Negative Bit
SRCX         = SR<4>,         ! Extend Bit
SRMASK       = SR<10:8>,      ! Interrupt Mask
FCSPACE      = FC<1:0>,       ! Memory Access Address Space
FCMODE       = FC<2>,         ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,      ! PC Low Word
PCHI         = PC<31:16>,     ! PC High Word
D0LWORD      = D[0]<15:0>,    ! D[0] Low Word
D1LWORD      = D[1]<15:0>,    ! D[1] Low Word
D2LWORD      = D[2]<15:0>,    ! D[2] Low Word
D3LWORD      = D[3]<15:0>,    ! D[3] Low Word
D4LWORD      = D[4]<15:0>,    ! D[4] Low Word
D5LWORD      = D[5]<15:0>,    ! D[5] Low Word
D6LWORD      = D[6]<15:0>,    ! D[6] Low Word
D7LWORD      = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***************************************************************************/
/*                                                                         */
/*                     16K 16-Bit Word Internal Memory                     */
/*                                                                         */
/***************************************************************************/

M[0:32767]<7:0>;

macro

/***************************************************************************/
/*                                                                         */
/*                    Logic Level Macros                                   */
/*                                                                        */
/************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/************************************************************************/
/*                                                                        */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                            */
/*                                                                        */
/************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100c] = 0xff;               ! Place Memory Locations Following The
        M[0x100d] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*      Routine Initialization Per Hamby and Guillory           */
        /*                                                               */
        /*****************************************************************/
        M[0x2000] = 0x55;               ! Initialize Memory Location
        M[0x2001] = 0x55;               ! 2000 Hex To 5555 Hex
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;                  ! Store Data At Hex 2000
        D[2] = 0x2000;                  ! D[2] Will Reset A[1]
        M[0x2002] = 0xaa;               ! Data Will Also Be Moved
        M[0x2003] = 0xaa;
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )
```

```
/************************************************************************/
/*                                                                      */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary    */
/*  to retrieve modeled instructions for simulation and analysis. It    */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/************************************************************************/

fetch_initial_instruction :=
      (

      /*****************************************************************/

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 0
      RW = hi;                              ! Memory Read
      ADENABLE = lo;                        ! Disable Address Bus Buffer
      DBENABLE = lo;                        ! Disable Data Bus Buffer
      IABUS = PC;                           ! Place PC On Internal Address
                                            ! Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2 Of
      PHI2 = hi;                            ! Clock Cycle 0
      ADENABLE = hi;                        ! Enable Address Bus Buffer
      EXABUF = IABUS;                       ! Gate Internal Address Bus
                                            ! Into External Address Buffer
      FCMODE = SRMODE;                      ! User Mode
      FCSPACE = 2;                          ! Accessing Program
      next;                                 ! Execute Impending Assignments
      ABUS = EXABUF;                        ! Address Placed On Bus(Added)
      next;                                 ! Execute Pending Assignments

      /*****************************************************************/
      T = 1;                                ! Clock Cycle 1
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 1
      ASN = lo;                             ! Assert Address Strobe
      LDSN = lo;                            ! Assert Lower Data Strobe
      UDSN = lo;                            ! Assert Upper Data Strobe
      DBENABLE = hi;                        ! Enable Data Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 1
      next;                                 ! Execute Pending Assignments

      /*****************************************************************/
      T = 2;                                ! Clock Cycle 2
      next;                                 ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/**************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/**************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
```

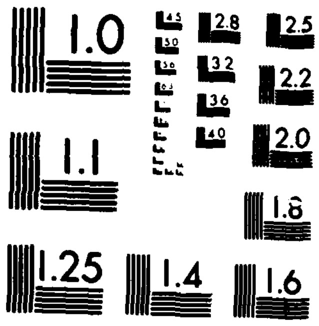
```
    LDSN = hi;                          ! Deactivate Lower Data Strobe
    UDSN = hi;                          ! Deactivate Upper Data Strobe
    IR = PFR;                           ! Contents Of Prefetch Register
                                        ! Are Placed Into Instruction
                                        ! Register
    DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
    PC = PC + 4;                        ! Increment Program Counter
    next;                               ! Execute Pending Assignments
    T = 0                               ! Reset Clock Cycle Counter
    )

movel :=                                 ! MOVE.L D2,A1
    (

    /****************************************************************/

    PHI1 = hi;                          ! Phase 1 Of
    PHI2 = lo;                          ! Clock Cycle 0
    ADENABLE = lo;                      ! Disable Address Bus
    DBENABLE = lo;                      ! Disable Data Bus
    DBUS = 0xffff;                      ! Place Data Bus In High Impedance
    RW = hi;                            ! Memory Read
    IABUS = PC;                         ! Place PC On Internal Address
                                        ! Bus
    IDBUS = D[2];                    ! Place Data From D[2] Onto
                                        ! Internal Data Bus
    next;                               ! Execute Pending Assignments

    PHI1 = lo;                          ! Phase 2 Of
    PHI2 = hi;                          ! Clock Cycle 0
    ADENABLE = hi;                      ! Enable Address Bus Buffer
    EXABUF = IABUS;                     ! Gate Internal Address Bus
                                        ! Into External Address Buffer
    FCMODE = SRMODE;                    ! User Mode
    FCSPACE = 2;                        ! Accessing Program
    SRCARRY = lo;                       ! Clear Status Register Carry Bit
    SROVER = lo;                        ! Clear Status Register Overflow Bit
    SRZERO = lo;                        ! Clear Status Register Zero Bit
    SRNEG  = lo;                        ! Clear Status Register Negative Bit
    A[1] = IDBUS;                    ! Place Data From Internal Data Bus
                                        ! Into A[1]
    next;                               ! Execute Impending Assignments
    ABUS = EXABUF;                      ! Address Placed On Bus(Added)
    next;                               ! Execute Pending Assignments

    /****************************************************************/
    T = 1;                              ! Clock Cycle 1
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1 Of
    PHI2 = lo;                          ! Clock Cycle 1
    ASN = lo;                           ! Assert Address Strobe
```

```
LDSN = lo;                               ! Assert Lower Data Strobe
UDSN = lo;                               ! Assert Upper Data Strobe
DBENABLE = hi;                           ! Enable Data Bus
if A[1] eql 0                          ! Set Status Register Zero Bit
   SRZERO = hi;                          ! If Moved Data Is Zero
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 1
if A[1]<31>                              ! Set Status Register Negative
   SRNEG = hi;                           ! Bit If Moved Data Is Negative
next;                                    ! Execute Pending Assignments

/************************************************************************/
T = 2;                                   ! Clock Cycle 2
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 2
      next;                              ! Execute Assignments

      /*****************************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*****************************************************************/
      T = 2                              ! Return To  Phase 2
                                         ! Of Clock Cycle 2
      );
      next;                              ! Execute Impending Assignments

/************************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
```

```
        next;                                   ! Execute Pending Assignments

        /*****************************************************************/
        T = 4;                                  ! Clock Cycle 4
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 4
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )


andi :=                                         ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                             ! Effect Of Instruction
       IR<15:8> = M[PC];                        ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                    ! Is To Set Status Register
          PC = PC + 2;                            ! Increment Program Counter
       T = 5;                                   ! Supervisor Bit To User
       next;                                    ! Mode
       T = 0                                    ! Requires 6 Clock Cycles
       )

move :=                                         ! MOVE.W (A1)+,D6
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        ABUS = 0xffffff;                        ! Address Bus High Impedanced
        DBENABLE = lo;                          ! Disable Data Bus Buffer
```

AD-A164 257 THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. C A BAXLEY 3/5

UNCLASSIFIED DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 NL



```
IABUS<31:1> = PC<31:1>;                    ! Place PC On Internal Address
                                           ! Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2 Of
PHI2 = hi;                                 ! Clock Cycle 0
ADENABLE = hi;                             ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                      ! Gate Internal Address Bus
                                           ! Into External Address Buffer
FCMODE = SRMODE;                           ! User Mode
FCSPACE = 2;                               ! Accessing Program
ABUS = IABUS<23:1>;                        ! Place PC On Address Bus
next;                                      ! Execute Impending Assignments

/***************************************************************/
T = 1;                                     ! Clock Cycle 1
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 1
ASN = lo;                                  ! Assert Address Strobe
LDSN = lo;                                 ! Assert Lower Data Strobe
UDSN = lo;                                 ! Assert Upper Data Strobe
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 1
next;                                      ! Execute Pending Assignments

/***************************************************************/
T = 2;                                     ! Clock Cycle 2
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 2
while DTACKN eql hi                        ! Wait For Memory To Place
      (                                    ! Data On The Bus
      next;                                ! Execute Impending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 2
      next;                                ! Execute Assignments

      /*********************************************************/
      T = 3;                               ! Clock Cycle 3
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
      DTACKN = lo;                         ! Asserts DTACKN(Added)
```

```
          next;                                   ! Execute Pending Assignments

          /*************************************************************/
          T = 2                                   ! Return To  Phase 2
                                                  ! Of Clock Cycle 2
          );
          next;                                   ! Execute Impending Assignments

     /*****************************************************************/
     T = 3;                                       ! Clock Cycle 3
     next;                                        ! Execute Assignment

     PHI1 = lo;                                   ! Phase 2
     PHI2 = hi;                                   ! Of Clock Cycle 3
     EXDBUF = DBUS;                               ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
     next;                                        ! Execute Pending Assignments

     /*****************************************************************/
     T = 4;                                       ! Clock Cycle 4
     next;                                        ! Execute Assignment

     PHI1 = hi;                                   ! Phase 1
     PHI2 = lo;                                   ! Of Clock Cycle 4
     PFR = EXDBUF;                                ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Prefetch Register
     next;                                        ! Execute Pending Assignments

     PHI1 = lo;                                   ! Phase 2
     PHI2 = hi;                                   ! Of Clock Cycle 4
     ASN = hi;                                    ! Deactivate Address Strobe
     LDSN = hi;                                   ! Deactivate Lower Data Strobe
     UDSN = hi;                                   ! Deactivate Upper Data Strobe
     PC = PC + 2;                                 ! Increment Program Counter
                                                  ! Are Placed Into Instruction
                                                  ! Register
     DTACKN = hi;                                 ! Deactivate Data Transfer(Added)
                                                  ! Acknowledge
     next;

     /*****************************************************************/
     T = 5;                                       ! Clock Cycle 5
     next;                                        ! Execute Previous Assignment

     PHI1 = hi;                                   ! Phase 1 Of
     PHI2 = lo;                                   ! Clock Cycle 5
     DBUS = 0xffff;                               ! Place Data Bus In High Impedance
     RW = hi;                                     ! Memory Read
     ADENABLE = lo;                               ! Disable Address Bus Buffer
     ABUS = 0xffffff;                             ! Address Bus High Impedanced
     DBENABLE = lo;                               ! Disable Data Bus Buffer
```

```
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Data
SRCARRY = lo;                           ! Clear Status Register Carry Bit
SROVER = lo;                            ! Clear Status Register Overflow Bit
SRZERO = lo;                            ! Clear Status Register Zero Bit
SRNEG  = lo;                            ! Clear Status Register Negative Bit
ABUS = IABUS<23:1>;                     ! Place PC On Address Bus (Added)
next;                                   ! Execute Impending Assignments

/******************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 7
      next;                             ! Execute Assignments

      /************************************************************/
      T = 8;                            ! Clock Cycle 8
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
```

```
          PHI2 = lo;                          ! Of Clock Cycle 8
          DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
          DTACKN = lo;                        ! Asserts DTACKN(Added)
          next;                               ! Execute Pending Assignments

          /*****************************************************/
          T = 7                               ! Return To  Phase 2
                                              ! Of Clock Cycle 7
          );
          next;                               ! Execute Impending Assignments

      /************************************************************/
      T = 8;                                  ! Clock Cycle 8
      next;                                   ! Execute Assignment

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 8
      EXDBUF = DBUS;                          ! Instruction On Data Bus
                                              ! Is Placed In External Data
                                              ! Bus Buffer
      A[1] = A[1] + 2;                        ! Increment A[1]
      next;                                   ! Execute Pending Assignments

      /************************************************************/
      T = 9;                                  ! Clock Cycle 9
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 9
      IDBUS = EXDBUF;
      if EXDBUF eql 0                         ! Set Status Register
         SRZERO = hi;                         ! Bits As Appropriate
      if EXDBUF<15> eql 1
         SRNEG = hi;
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 9
      if IR eql 0x3c19
         D[6] = IDBUS                         ! Place Contents Of Internal
      else                                    ! Data Bus Into D[6]/D[7]
         D[7] = IDBUS;
      ASN = hi;                               ! Deactivate Address Strobe
      LDSN = hi;                              ! Deactivate Lower Data Strobe
      UDSN = hi;                              ! Deactivate Upper Data Strobe
      IR = PFR;                               ! Contents Of Prefetch Register
                                              ! Are Placed Into Instruction
                                              ! Register
      DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                              ! Acknowledge
      next;                                   ! Execute Impending Assignments
      T = 0                                   ! Reset Clock Cycle Counter
```

```
        )

jmp :=                                                  ! JMP (A0)
        (

        /*************************************************************/

        PHI1 = hi;                                      ! Phase 1 Of
        PHI2 = lo;                                      ! Clock Cycle 0
        DBUS = 0xffff;                                  ! Place Data Bus In A High Impedance
        RW = hi;                                        ! Memory Read
        ADENABLE = lo;                                  ! Disable Address Bus Buffer
        DBENABLE = lo;                                  ! Disable Data Bus Buffer
        IABUS = PC;                                     ! Place PC On Internal Address
                                                        ! Bus
        next;                                           ! Execute Pending Assignments

        PHI1 = lo;                                      ! Phase 2 Of
        PHI2 = hi;                                      ! Clock Cycle 0
        ADENABLE = hi;                                  ! Enable Address Bus Buffer
        EXABUF = IABUS;                                 ! Gate Internal Address Bus
                                                        ! Into External Address Buffer
        FCMODE = SRMODE;                                ! User Mode
        FCSPACE = 2;                                    ! Accessing Program
        next;                                           ! Execute Pending Assignments
        ABUS = EXABUF;                                  ! Address Placed On Bus(Added)
        next;                                           ! Execute Pending Assignments

        /*************************************************************/
        T = 1;                                          ! Clock Cycle 1
        next;                                           ! Execute Assignment

        PHI1 = hi;                                      ! Phase 1 Of
        PHI2 = lo;                                      ! Clock Cycle 1
        ASN = lo;                                       ! Assert Address Strobe
        LDSN = lo;                                      ! Assert Lower Data Strobe
        UDSN = lo;                                      ! Assert Upper Data Strobe
        IABUS = A[0];                                   ! Move Jump Address From A[0]
                                                        ! To Internal Address Buffer
        DBENABLE = hi;                                  ! Enable Data Bus
        next;                                           ! Execute Pending Assignments

        PHI1 = lo;                                      ! Phase 2
        PHI2 = hi;                                      ! Of Clock Cycle 1
        PC = IABUS;                                     ! Place Jump Address Into Program
                                                        ! Counter
        next;

        /*************************************************************/
        T = 2;                                          ! Clock Cycle 2
        next;                                           ! Execute Assignment

        PHI1 = hi;                                      ! Phase 1
```

```
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
     (                                   ! Data On The Bus
     next;                               ! Execute Impending Assignments

     PHI1 = lo;                          ! Phase 2
     PHI2 = hi;                          ! Of Clock Cycle 2
     next;                               ! Execute Assignments

     /************************************************************/
     T = 3;                              ! Clock Cycle 3
     next;                               ! Execute Assignment

     PHI1 = hi;                          ! Phase 1
     PHI2 = lo;                          ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
     DTACKN = lo;                        ! Asserts DTACKN(Added)
     next;                               ! Execute Pending Assignments

     /************************************************************/
     T = 2                               ! Return To Phase 2
                                         ! Of Clock Cycle 2
     );
     next;                               ! Execute Impending Assignments

/****************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/****************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
next;
PFR = EXDBUF;                            ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 4
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
```

```
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments
```

```
        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 7
        next;                                   ! Execute Assignments

        /*******************************************************/
        T = 8;                                  ! Clock Cycle 8
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 8
        DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
        DTACKN = lo;                            ! Asserts DTACKN(Added)
        next;                                   ! Execute Pending Assignments

        /*******************************************************/
        T = 7                                  ! Return To Phase 2
                                               ! Of Clock Cycle 7
        );
        next;                                  ! Execute Impending Assignments

    /*************************************************************/
    T = 8;                                      ! Clock Cycle 8
    next;                                       ! Execute Assignment

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 8
    EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
    next;                                       ! Execute Pending Assignments

    /*************************************************************/
    T = 9;                                      ! Clock Cycle 9
    next;                                       ! Execute Assignment

    PHI1 = hi;                                  ! Phase 1
    PHI2 = lo;                                  ! Of Clock Cycle 9
    PFR = EXDBUF;                               ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
    next;                                       ! Execute Pending Assignments

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 9
    ASN = hi;                                   ! Deactivate Address Strobe
    LDSN = hi;                                  ! Deactivate Lower Data Strobe
    UDSN = hi;                                  ! Deactivate Upper Data Strobe
    PC = PC + 2;                                ! Increment Program Counter
    IR = PFR;                                   ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
    DTACKN = hi;                                ! Deactivate Data Transfer
```

```
                                                    ! Acknowledge(Added)
            next;                                   ! Execute Pending Assignments
            T = 0                                   ! Reset Clock Cycle Counter
            )

      decode_execute_prefetch :=
                              (
                              case IR
                                        0x2242: movel   ! MOVE.L D2,A1
                                  0x3c19,0x3e19: move    ! MOVE.W (A1)+,D6
                                        0x027c: andi    ! AND.W #$DFFF,SR
                                        047320: jmp     ! JMP (A0)
                              esac
                              )

      main :=
           (
           power_on_initialize;
           fetch_initial_instruction;
           while READY eql hi
                (
                decode_execute_prefetch
                )
           )
```

```
/***********************************************************************/
/*                                                                     */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W -(A1),D4 INSTRUCTION       */
/*                                                                     */
/***********************************************************************/

/***********************************************************************/
/*                                                                     */
/*                     Structure Declarations                          */
/*                                                                     */
/***********************************************************************/

state

/***********************************************************************/
/*                                                                     */
/*                 M68000 Programming Registers                        */
/*                                                                     */
/***********************************************************************/

D[0:7]<31:0>,              ! 8 Data Registers
A[0:6]<31:0>,              ! 7 Address Registers
UA7<31:0>,                 ! User Stack Pointer
SA7<31:0>,                 ! System Stack Pointer
PC<31:0>,                  ! Program Counter
SR<15:0>,                  ! Status Register

/***********************************************************************/
/*                                                                     */
/*                 Temporary Internal Registers                        */
/*                                                                     */
/***********************************************************************/

PFR<15:0>,                 ! Prefetch Register
IR<15:0>,                  ! Instruction Register
FC<2:0>,                   ! Function Code Register
EXDBUF<15:0>,              ! External Data Bus Buffer Register
EXABUF<23:1>,              ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,             ! ALU Buffer 1
ALUBUF2<31:0>,             ! ALU Buffer 2
DTEMP<15:0>,               ! Temporary Data Storage
DISREG<31:0>,              ! Temporary Displacement Storage
SRTEMP<15:0>,              ! Temporary Status Register Storage
                           ! (Exception Processing)
IRTEMP<15:0>,              ! Temporary Instruction Register Storage
                           ! (Exception Processing)
TEMPADR<31:0>,             ! Temporary Cycle Address Storage
                           ! (Exception Processing)
ACTYPE<15:0>,              ! Temporary Access Type Storage
                           ! (Exception Processing)
VECADR<23:0>,              ! Temporary Vector Address Storage
                           ! (Exception Processing)
```

```
HANADR<31:0>,                   ! Temporary Address Storage For
                                ! Exception Handler Routine
T<7:0>,                         ! Clock Cycle Counter
RESET,                       ! Reset Flip-Flop
HALT,                        ! Halt Flip-Flop
RW,                          ! Read/Write Flip-Flop
ADENABLE,                    ! Address Bus Buffer Enable
DBENABLE,                    ! Data Bus Buffer Enable
ASN,                         ! Address Strobe Flip-Flop
LDSN,                        ! Lower Data Strobe Flip-Flop
UDSN,                        ! Upper Data Strobe Flip-Flop
DTACKN,                      ! Data Transfer Acknowledge Flip-Flop
COUT,                        ! Carry Flip-Flop
EXCEPT,                      ! Exception Processing Flip-Flop
READY,                       ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                           */
/*                                                                    */
/*           1) CDL decoder structure nonexistent in ISP' and un-     */
/*      necessary for model. Eliminated.                              */
/*           2) Multi-phase clock structure nonexistent in ISP'.      */
/*      Operations on registers will provide its equivalent.          */
/*           3) Switch structure nonexistent in ISP'. Operation on a  */
/*      register will provide its equivalent.                         */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*           5) The memory word length was reduced from 16 to 8 bit   */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,                    ! Internal Address Bus
IDBUS<31:0>,                    ! Internal Data Bus
twait<4:0>,                     ! Wait State Counter
SWITCH,                      ! Power Switch
PHI1,                        ! Phase 1 Of Two-Phase Clock
PHI2;                        ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                     ! External Data Bus
```

```
ABUS<23:1>;                     ! External Address Bus(changed)

format

/***********************************************************************/
/*                                                                     */
/*                       Register Subfields                            */
/*                                                                     */
/***********************************************************************/

PCADDR       = PC<23:0>,        ! Program Counter Address Field
SRTRACE      = SR<15>,          ! Trace Bit
SRMODE       = SR<13>,          ! Mode Selection Bit
SRCARRY      = SR<0>,           ! Carry Bit
SROVER       = SR<1>,           ! Overflow Bit
SRZERO       = SR<2>,           ! Zero Bit
SRNEG        = SR<3>,           ! Negative Bit
SREX         = SR<4>,           ! Extend Bit
SRMASK       = SR<10:8>,        ! Interrupt Mask
FCSPACE      = FC<1:0>,         ! Memory Access Address Space
FCMODE       = FC<2>,           ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,        ! PC Low Word
PCHI         = PC<31:16>,       ! PC High Word
D0LWORD      = D[0]<15:0>,      ! D[0] Low Word
D1LWORD      = D[1]<15:0>,      ! D[1] Low Word
D2LWORD      = D[2]<15:0>,      ! D[2] Low Word
D3LWORD      = D[3]<15:0>,      ! D[3] Low Word
D4LWORD      = D[4]<15:0>,      ! D[4] Low Word
D5LWORD      = D[5]<15:0>,      ! D[5] Low Word
D6LWORD      = D[6]<15:0>,      ! D[6] Low Word
D7LWORD      = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***********************************************************************/
/*                                                                     */
/*                     16K 16-Bit Word Internal Memory                 */
/*                                                                     */
/***********************************************************************/

M[0:32767]<7:0>;

macro

/***********************************************************************/
/*                                                                     */
/*                       Logic Level Macros                            */
/*                                                                      */
/************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/************************************************************************/
/*                                                                      */
/*  Power On and Initialization. This process was not modeled but is   */
/*  added to initialize signals and registers.                          */
/*                                                                      */
/************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100c] = 0xff;                ! Place Memory Locations Following The
        M[0x100d] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /***************************************************************/
        /*                                                             */
        /*     Routine Initialization Per Hamby and Guillory           */
        /*                                                             */
        /***************************************************************/
        M[0x2006] = 0x55;               ! Data To Be Moved
        M[0x2007] = 0x55;
        M[0x2004] = 0xaa;
        M[0x2005] = 0xaa;
        D[2] = 0x2008;                  ! Place 2008 Into D[2]
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        A[1] = 0x2008;               ! Store Data At This Address
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )
```

```
/**********************************************************************/
/*                                                                    */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary  */
/*  to retrieve modeled instructions for simulation and analysis. It  */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                    */
/*                                                                    */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /**************************************************************/

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 0
      RW = hi;                            ! Memory Read
      ADENABLE = lo;                      ! Disable Address Bus Buffer
      DBENABLE = lo;                      ! Disable Data Bus Buffer
      IABUS = PC;                         ! Place PC On Internal Address
                                          ! Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2 Of
      PHI2 = hi;                          ! Clock Cycle 0
      ADENABLE = hi;                      ! Enable Address Bus Buffer
      EXABUF = IABUS;                     ! Gate Internal Address Bus
                                          ! Into External Address Buffer
      FCMODE = SRMODE;                    ! User Mode
      FCSPACE = 2;                        ! Accessing Program
      next;                               ! Execute Impending Assignments
      ABUS = EXABUF;                      ! Address Placed On Bus(Added)
      next;                               ! Execute Pending Assignments

      /**************************************************************/
      T = 1;                              ! Clock Cycle 1
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 1
      ASN = lo;                           ! Assert Address Strobe
      LDSN = lo;                          ! Assert Lower Data Strobe
      UDSN = lo;                          ! Assert Upper Data Strobe
      DBENABLE = hi;                      ! Enable Data Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 1
      next;                               ! Execute Pending Assignments

      /**************************************************************/
      T = 2;                              ! Clock Cycle 2
      next;                               ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /************************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /************************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
```

```
       LDSN = hi;                              ! Deactivate Lower Data Strobe
       UDSN = hi;                              ! Deactivate Upper Data Strobe
       IR = PFR;                               ! Contents Of Prefetch Register
                                               ! Are Placed Into Instruction
                                               ! Register
       DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                               ! Acknowledge
       PC = PC + 4;                            ! Increment Program Counter
       next;                                   ! Execute Pending Assignments
       T = 0                                   ! Reset Clock Cycle Counter
       )

andi :=                                        ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                            ! Effect Of Instruction
       IR<15:8> = M[PC];                       ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                   ! Is To Set Status Register
         PC = PC + 2;                            ! Increment Program Counter
       T = 5;                                  ! Supervisor Bit To User
       next;                                   ! Mode
       T = 0                                   ! Requires 6 Clock Cycles
       )

move :=                                        ! MOVE.W -(A1),D4
        (

       /*****************************************************************/

       PHI1 = hi;                              ! Phase 1 Of
       PHI2 = lo;                              ! Clock Cycle 0
       DBUS = 0xffff;                          ! Place Data Bus In High Impedance
       RW = hi;                                ! Memory Read
       ADENABLE = lo;                          ! Disable Address Bus Buffer
       ABUS = 0xffffff;                        ! Address Bus High Impedanced
       DBENABLE = lo;                          ! Disable Data Bus Buffer
       IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                               ! Bus
       next;                                   ! Execute Pending Assignments

       PHI1 = lo;                              ! Phase 2 Of
       PHI2 = hi;                              ! Clock Cycle 0
       ADENABLE = hi;                          ! Enable Address Bus Buffer
       EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                               ! Into External Address Buffer
       FCMODE = SRMODE;                        ! User Mode
       FCSPACE = 2;                            ! Accessing Program
       ABUS = IABUS<23:1>;                     ! Address Placed On Bus
       next;                                   ! Execute Impending Assignments

       /*****************************************************************/
       T = 1;                                  ! Clock Cycle 1
       next;                                   ! Execute Assignment
```

```
PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 1
ASN = lo;                                  ! Assert Address Strobe
LDSN = lo;                                 ! Assert Lower Data Strobe
UDSN = lo;                                 ! Assert Upper Data Strobe
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 1
next;                                      ! Execute Pending Assignments

/****************************************************************/
T = 2;                                     ! Clock Cycle 2
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 2
while DTACKN eql hi                        ! Wait For Memory To Place
    (                                      ! Data On The Bus
    next;                                  ! Execute Impending Assignments

    PHI1 = lo;                             ! Phase 2
    PHI2 = hi;                             ! Of Clock Cycle 2
    next;                                  ! Execute Assignments

    /************************************************************/
    T = 3;                                 ! Clock Cycle 3
    next;                                  ! Execute Assignment

    PHI1 = hi;                             ! Phase 1
    PHI2 = lo;                             ! Of Clock Cycle 3
    DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
    DTACKN = lo;                           ! Asserts DTACKN(Added)
    next;                                  ! Execute Pending Assignments

    /************************************************************/
    T = 2                                  ! Return To  Phase 2
                                           ! Of Clock Cycle 2
    );
    next;                                  ! Execute Impending Assignments

/****************************************************************/
T = 3;                                     ! Clock Cycle 3
next;                                      ! Execute Assignment

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 3
EXDBUF = DBUS;                             ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
```

```
next;                                      ! Execute Pending Assignments

/****************************************************************/
T = 4;                                     ! Clock Cycle 4
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 4
PFR = EXDBUF;                              ! Prefetch Register Gets External
                                           ! Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 4
ASN = hi;                                  ! Deactivate Address Strobe
LDSN = hi;                                 ! Deactivate Lower Data Strobe
UDSN = hi;                                 ! Deactivate Upper Data Strobe
                                           ! Are Placed Into Instruction
                                           ! Register
PC = PC + 2;                               ! Increment Program Counter
DTACKN = hi;                               ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
next;

/****************************************************************/
T = 5;                                     ! Clock Cycle 5
next;                                      ! Execute Previous Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 5
A[1] = A[1] - 2;                           ! Decrement A[1]
ABUS = 0xffffff;                           ! Address Bus High Impedanced
DBUS = 0xffff;                             ! Data Bus High Impedanced
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2 Of
PHI2 = hi;                                 ! Clock Cycle 5
next;                                      ! Into External Address Buffer

/****************************************************************/
T = 6;                                     ! Clock Cycle 6
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 6
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 6
next;                                      ! Execute Pending Assignments

/****************************************************************/
T = 7;                                     ! Clock Cycle 7
```

```
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 7
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 7
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Data
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                     ! Place Address On Bus
SRCARRY = 0;                            ! Initialize Status Register
SROVER = 0;                             ! Condition Bits
SRZERO = 0;
SRNEG = 0;
next;                                   ! Into External Address Buffer

/*****************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 8
UDSN = lo;                              ! Activate Upper And
LDSN = lo;                              ! Lower Data Strobes
ASN = lo;                               ! Assert Address Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                  ! Clock Cycle 9
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 9
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 9
      next;                             ! Execute Assignments
```

```
          /***********************************************************/
          T = 10;                              ! Clock Cycle 10
          next;                                ! Execute Assignment

          PHI1 = hi;                           ! Phase 1
          PHI2 = lo;                           ! Of Clock Cycle 10
          DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
          DTACKN = lo;                         ! Asserts DTACKN(Added)
          next;                                ! Execute Pending Assignments

          /***********************************************************/
          T = 9                                ! Return To  Phase 2
                                               ! Of Clock Cycle 9
          );
          next;                                ! Execute Impending Assignments

      /***************************************************************/
      T = 10;                                  ! Clock Cycle 10
      next;                                    ! Execute Assignment

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 10
      EXDBUF = DBUS;                           ! Instruction On Data Bus
                                               ! Is Placed In External Data
                                               ! Bus Buffer
      next;                                    ! Execute Pending Assignments

      /***************************************************************/
      T = 11;                                  ! Clock Cycle 11
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 11
      IDBUS = EXDBUF;
      if EXDBUF eql 0                          ! Set Condition Code Bits
         SRZERO = hi;                          ! As Appropriate
      if EXDBUF<15>
         SRNEG = hi;
      next;                                    ! Execute Pending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 11
      if IR eql 0x3821                         ! Place Value In Either
         D[4] = IDBUS                          ! D[4] Or D[3]
      else                                     ! Depending On Instruction
         D[3] = IDBUS;
      ASN = hi;                                ! Deactivate Address Strobe
      LDSN = hi;                               ! Deactivate Lower Data Strobe
      UDSN = hi;                               ! Deactivate Upper Data Strobe
      DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                               ! Acknowledge
```

```
        IR = PFR;
        next;                                   ! Execute Pending Assignments

        T = 0
        )

movel   :=                                      ! MOVE.L D2,A1
        (

        /****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        ADENABLE = lo;                          ! Disable Address Bus
        DBENABLE = lo;                          ! Disable Data Bus
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        IDBUS = D[2];                       ! Place Data From D[2] Onto
                                                ! Internal Data Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        SRCARRY = lo;                           ! Clear Status Register Carry Bit
        SROVER = lo;                            ! Clear Status Register Overflow Bit
        SRZERO = lo;                            ! Clear Status Register Zero Bit
        SRNEG  = lo;                            ! Clear Status Register Negative Bit
        A[1] = IDBUS;                       ! Place Data From Internal Data Bus
                                                ! Into A[1]
        next;                                   ! Execute Impending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        DBENABLE = hi;                          ! Enable Data Bus
        if A[1] eql 0                       ! Set Status Register Zero Bit
           SRZERO = hi;                         ! If Moved Data Is Zero
        next;                                   ! Execute Pending Assignments
```

```
PHI1 = lo;                                    ! Phase 2
PHI2 = hi;                                    ! Of Clock Cycle 1
if A[1]<31>                                   ! Set Status Register Negative
   SRNEG = hi;                                ! Bit If Moved Data Is Negative
next;                                         ! Execute Pending Assignments

/*************************************************************************/
T = 2;                                        ! Clock Cycle 2
next;                                         ! Execute Assignment

PHI1 = hi;                                    ! Phase 1
PHI2 = lo;                                    ! Of Clock Cycle 2
while DTACKN eql hi                           ! Wait For Memory To Place
      (                                       ! Data On The Bus
      next;                                   ! Execute Impending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 2
      next;                                   ! Execute Assignments

      /******************************************************************/
      T = 3;                                  ! Clock Cycle 3
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
      DTACKN = lo;                            ! Asserts DTACKN(Added)
      next;                                   ! Execute Pending Assignments

      /******************************************************************/
      T = 2                                   ! Return To  Phase 2
                                              ! Of Clock Cycle 2
      );
      next;                                   ! Execute Impending Assignments

/*************************************************************************/
T = 3;                                        ! Clock Cycle 3
next;                                         ! Execute Assignment

PHI1 = lo;                                    ! Phase 2
PHI2 = hi;                                    ! Of Clock Cycle 3
EXDBUF = DBUS;                                ! Instruction On Data Bus
                                              ! Is Placed In External Data
                                              ! Bus Buffer
next;                                         ! Execute Pending Assignments

/*************************************************************************/
T = 4;                                        ! Clock Cycle 4
next;                                         ! Execute Assignment
```

```
        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 4
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    jmp :=                                      ! JMP (A0)
        (

        /****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
```

```
PHI2 = lo;                                   ! Clock Cycle 1
ASN = lo;                                    ! Assert Address Strobe
LDSN = lo;                                   ! Assert Lower Data Strobe
UDSN = lo;                                   ! Assert Upper Data Strobe
IABUS = A[0];                                ! Move Jump Address From A[0]
                                             ! To Internal Address Buffer
DBENABLE = hi;                               ! Enable Data Bus
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 1
PC = IABUS;                                  ! Place Jump Address Into Program
                                             ! Counter
next;

/*****************************************************************/
T = 2;                                       ! Clock Cycle 2
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1
PHI2 = lo;                                   ! Of Clock Cycle 2
while DTACKN eql hi                          ! Wait For Memory To Place
      (                                      ! Data On The Bus
      next;                                  ! Execute Impending Assignments

      PHI1 = lo;                             ! Phase 2
      PHI2 = hi;                             ! Of Clock Cycle 2
      next;                                  ! Execute Assignments

      /**********************************************************/
      T = 3;                                 ! Clock Cycle 3
      next;                                  ! Execute Assignment

      PHI1 = hi;                             ! Phase 1
      PHI2 = lo;                             ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
      DTACKN = lo;                           ! Asserts DTACKN(Added)
      next;                                  ! Execute Pending Assignments

      /**********************************************************/
      T = 2                                  ! Return To Phase 2
                                             ! Of Clock Cycle 2
      );
      next;                                  ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                       ! Clock Cycle 3
next;                                        ! Execute Assignment

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 3
EXDBUF = DBUS;                               ! Instruction On Data Bus
```

```
                                      ! Is Placed In External Data
                                      ! Bus Buffer
next;                                 ! Execute Pending Assignments

/************************************************************************/
T = 4;                                ! Clock Cycle 4
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 4
next;
PFR = EXDBUF;                         ! The Contents Of The External
                                      ! Data Bus Buffer Are Placed
                                      ! In Prefetch Register
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 4
ASN = hi;                             ! Deactivate Address Strobe
LDSN = hi;                            ! Deactivate Lower Data Strobe
UDSN = hi;                            ! Deactivate Upper Data Strobe
DTACKN = hi;                          ! Deactivate Data Transfer
                                      ! Acknowledge(Added)
next;
/************************************************************************/
T = 5;                                ! Clock Cycle 5
next;                                 ! Execute Previous Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 5
RW = hi;                              ! Memory Read
ADENABLE = lo;                        ! Disable Address Bus Buffer
DBENABLE = lo;                        ! Disable Data Bus Buffer
IABUS = PC;                           ! Place PC On Internal Address
                                      ! Bus
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2 Of
PHI2 = hi;                            ! Clock Cycle 5
ADENABLE = hi;                        ! Enable Address Bus Buffer
FCMODE = SRMODE;                      ! User Mode
FCSPACE = 2;                          ! Accessing Program
EXABUF = IABUS;                       ! Gate Internal Address Bus
next;                                 ! Into External Address Buffer
ABUS = EXABUF;                        ! Address Placed On Bus(Added)
next;                                 ! Execute Pending Assignments

/************************************************************************/
T = 6;                                ! Clock Cycle 6
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 6
```

```
ASN = lo;                                   ! Assert Address Strobe
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*******************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
    (                                       ! Data On The Bus
    next;                                   ! Execute Impending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 7
    next;                                   ! Execute Assignments

    /***************************************************************/
    T = 8;                                  ! Clock Cycle 8
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 8
    DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
    DTACKN = lo;                            ! Asserts DTACKN(Added)
    next;                                   ! Execute Pending Assignments

    /***************************************************************/
    T = 7                                 ! Return To Phase 2
                                          ! Of Clock Cycle 7
    );
    next;                                 ! Execute Impending Assignments

/*******************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 8
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*******************************************************************/
```

```
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             0x3821,0x3621: move    ! MOVE.W -(A1),D4 [D3]
                                    0x027c: andi    ! AND.W #$DFFF,SR
                                    047320: jmp     ! JMP (A0)
                                    0x2242: movel   ! MOVE.L D2,A1
                        esac
                        )

main :=
     (
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
          (
          decode_execute_prefetch
          )
     )
```

```
/*********************************************************************/
/*                                                                   */
/*    MOTOROLA MC68000 MODEL OF THE MOVE.W 04(A1),D1 INSTRUCTION     */
/*                                                                   */
/*********************************************************************/

/*********************************************************************/
/*                                                                   */
/*                    Structure Declarations                         */
/*                                                                   */
/*********************************************************************/

state

/*********************************************************************/
/*                                                                   */
/*                M68000 Programming Registers                       */
/*                                                                   */
/*********************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/*********************************************************************/
/*                                                                   */
/*                Temporary Internal Registers                       */
/*                                                                   */
/*********************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/***********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                           */
/*                                                                    */
/*            1) CDL decoder structure nonexistent in ISP' and un-    */
/*      necessary for model. Eliminated.                              */
/*            2) Multi-phase clock structure nonexistent in ISP'.     */
/*      Operations on registers will provide its equivalent.          */
/*            3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                         */
/*            4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*            5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/***********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
twait<4:0>,                  ! Wait State Counter
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/***********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/***********************************************************************/

DBUS<15:0>,                  ! External Data Bus
```

```
ABUS<23:1>;                    ! External Address Bus(changed)

format

/***************************************************************************/
/*                                                                         */
/*                      Register Subfields                                 */
/*                                                                         */
/***************************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***************************************************************************/
/*                                                                         */
/*                      16K 16-Bit Word Internal Memory                    */
/*                                                                         */
/***************************************************************************/

M[0:32767]<7:0>;

macro

/***************************************************************************/
/*                                                                         */
/*                      Logic Level Macros                                 */
```

411

```
/*                                                                          */
/****************************************************************************/

lo    = 0 %,
hi    = 1 %,
off   = 0 %,
on    = 1 %,
clear = 0 %;

/****************************************************************************/
/*                                                                          */
/*  Power On and Initialization. This process was not modeled but is      */
/*  added to initialize signals and registers.                             */
/*                                                                          */
/****************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                ! Place Memory Locations Following The
        M[0x100f] = 0xff;                 ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*******************************************************************/
        /*                                                                 */
        /*      Routine Initialization Per Hamby and Guillory              */
        /*                                                                 */
        /*******************************************************************/
        M[0x2004] = 0x55;               ! Value To Be Moved
        M[0x2005] = 0x55;
        M[0x2008] = 0xaa;               ! Value To Be Moved
        M[0x2009] = 0xaa;
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;               ! Store Data At This Address
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/****************************************************************************/
```

```
/*                                                                          */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary        */
/*  to retrieve modeled instructions for simulation and analysis. It        */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory     */
/*  on page VI-15 of their thesis.                                          */
/*                                                                          */
/**************************************************************************/

fetch_initial_instruction :=
      (

      /******************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /******************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /******************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
     (                                  ! Data On The Bus
     next;                              ! Execute Impending Assignments

     PHI1 = lo;                         ! Phase 2
     PHI2 = hi;                         ! Of Clock Cycle 2
     next;                              ! Execute Assignments

     /*******************************************************/
     T = 3;                             ! Clock Cycle 3
     next;                              ! Execute Assignment

     PHI1 = hi;                         ! Phase 1
     PHI2 = lo;                         ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
     DTACKN = lo;                       ! Asserts DTACKN(Added)
     next;                              ! Execute Pending Assignments

     /*******************************************************/
     T = 2                              ! Return To  Phase 2
                                        ! Of Clock Cycle 2
     );
     next;                              ! Execute Impending Assignments

/***********************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/***********************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
```

```
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 4;                            ! Increment Program Counter
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

andi :=                                         ! AND.W #$DFFF,SR
        (
       SRMODE = lo;                             ! Effect Of Instruction
       IR<15:8> = M[PC];                        ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                    ! Is To Set Status Register
          PC = PC + 2;                            ! Increment Program Counter
       T = 5;                                   ! Supervisor Bit To User
       next;                                    ! Mode
       T = 0                                    ! Requires 6 Clock Cycles
       )

move :=                                         ! MOVE.W 4(A1),D1 [8(A1),D2]
        (

        /*******************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        ABUS = IABUS;                           ! Address Placed On Bus
        next;                                   ! Execute Impending Assignments

        /*******************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
```

```
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /**********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /**********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
DISREG = EXDBUF sxt 32;                  ! Store Displacement
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 4
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
UDSN = hi;                               ! Deactivate Upper Data Strobe
                                         ! Are Placed Into Instruction
                                         ! Register
PC = PC + 2;                             ! Increment Program Counter
DISREG = DISREG + A[1];                ! Add Address Register To Displacement
DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
next;

/*****************************************************************/
T = 5;                                   ! Clock Cycle 5
next;                                    ! Execute Previous Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 5
RW = hi;                                 ! Memory Read
ADENABLE = lo;                           ! Disable Address Bus Buffer
DBUS = 0xffff;                           ! Data Bus Returned To High
                                         ! Impedance State
DBENABLE = lo;                           ! Disable Data Bus Buffer
IABUS = PC;                              ! Place PC On Internal Address
                                         ! Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2 Of
PHI2 = hi;                               ! Clock Cycle 5
ADENABLE = hi;                           ! Enable Address Bus Buffer
FCMODE = SRMODE;                         ! User Mode
FCSPACE = 2;                             ! Accessing Data
EXABUF = IABUS;                          ! Gate Internal Address Bus
ABUS = IABUS;                            ! Place Address On Bus
next;                                    ! Into External Address Buffer

/*****************************************************************/
T = 6;                                   ! Clock Cycle 6
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 6
UDSN = lo;                               ! Activate Upper And
```

```
LDSN = lo;                                ! Lower Data Strobes
ASN = lo;                                 ! Assert Address Strobe
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 6
next;                                     ! Execute Pending Assignments

/**************************************************************/
T = 7;                                    ! Clock Cycle 7
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 7
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 7
      next;                               ! Execute Assignments

      /*******************************************************/
      T = 8;                              ! Clock Cycle 8
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
      DTACKN = lo;                        ! Asserts DTACKN(Added)
      next;                               ! Execute Pending Assignments

      /*******************************************************/
      T = 7                               ! Return To  Phase 2
                                          ! Of Clock Cycle 7
      );
      next;                               ! Execute Impending Assignments

/**************************************************************/
T = 8;                                    ! Clock Cycle 8
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 8
EXDBUF = DBUS;                            ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
next;                                     ! Execute Pending Assignments

/**************************************************************/
T = 9;                                    ! Clock Cycle 9
```

```
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 9
PFR = EXDBUF;                             ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 9
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
UDSN = hi;                                ! Deactivate Upper Data Strobe
PC = PC + 2;                              ! Increment PC
DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 10;                                   ! Clock Cycle 10
next;                                     ! Execute Previous Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 10
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
DBUS = 0xffff;                            ! Data Bus Returned To High
                                          ! Impedance State
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS = DISREG;                           ! Place DISREG[1] On Internal
                                          ! Address Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 10
ADENABLE = hi;                            ! Enable Address Bus Buffer
FCMODE = SRMODE;                          ! User Mode
FCSPACE = 1;                              ! Accessing Data
EXABUF = IABUS;                           ! Gate Internal Address Bus
ABUS = IABUS;                             ! Place Address On Bus
SRCARRY = 0;                              ! Initialize Status Register
SROVER = 0;                               ! Condition Bits
SRZERO = 0;
SRNEG = 0;
next;                                     ! Into External Address Buffer

/*****************************************************************/
T = 11;                                   ! Clock Cycle 11
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 11
```

```
UDSN = lo;                                 ! Activate Upper And
LDSN = lo;                                 ! Lower Data Strobes
ASN = lo;                                  ! Assert Address Strobe
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 11
next;                                      ! Execute Pending Assignments

/****************************************************************/
T = 12;                                     ! Clock Cycle 12
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 12
while DTACKN eql hi                        ! Wait For Memory To Place
      (                                    ! Data On The Bus
      next;                                ! Execute Impending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 12
      next;                                ! Execute Assignments

      /**********************************************************/
      T = 13;                               ! Clock Cycle 13
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 13
      DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
      DTACKN = lo;                         ! Asserts DTACKN(Added)
      next;                                ! Execute Pending Assignments

      /**********************************************************/
      T = 12                               ! Return To  Phase 2
                                           ! Of Clock Cycle 12
      );
      next;                                ! Execute Impending Assignments

/****************************************************************/
T = 13;                                     ! Clock Cycle 13
next;                                      ! Execute Assignment

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 13
EXDBUF = DBUS;                             ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
next;                                      ! Execute Pending Assignments

/****************************************************************/
```

```
        T = 14;                                     ! Clock Cycle 14
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1
        PHI2 = lo;                                  ! Of Clock Cycle 14
        IDBUS = EXDBUF;
        if EXDBUF eql 0                             ! Set Condition Code Bits
           SRZERO = hi;                             ! As Appropriate
        if EXDBUF<15>
           SRNEG = hi;
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 14
        if IR eql 0x3229                            ! Place Value In Either
           D[1] = IDBUS                             ! D[1] Or D[2]
        else                                        ! Depending On Instruction
           D[2] = IDBUS;
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                                    ! Acknowledge
        IR = PFR;
        next;                                       ! Execute Pending Assignments

        T = 0
        )

    jmp :=                                          ! JMP (A0)
        (

        /**********************************************************************/

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 0
        DBUS = 0xffff;                              ! Place Data Bus In A High Impedance
        RW = hi;                                    ! Memory Read
        ADENABLE = lo;                              ! Disable Address Bus Buffer
        DBENABLE = lo;                              ! Disable Data Bus Buffer
        IABUS = PC;                                 ! Place PC On Internal Address
                                                    ! Bus
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2 Of
        PHI2 = hi;                                  ! Clock Cycle 0
        ADENABLE = hi;                              ! Enable Address Bus Buffer
        EXABUF = IABUS;                             ! Gate Internal Address Bus
                                                    ! Into External Address Buffer
        FCMODE = SRMODE;                            ! User Mode
        FCSPACE = 2;                                ! Accessing Program
        next;                                       ! Execute Pending Assignments
        ABUS = EXABUF;                              ! Address Placed On Bus(Added)
```

```
next;                                 ! Execute Pending Assignments

/*************************************************************/
T = 1;                                ! Clock Cycle 1
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 1
ASN = lo;                             ! Assert Address Strobe
LDSN = lo;                            ! Assert Lower Data Strobe
UDSN = lo;                            ! Assert Upper Data Strobe
IABUS = A[0];                         ! Move Jump Address From A[0]
                                      ! To Internal Address Buffer
DBENABLE = hi;                        ! Enable Data Bus
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 1
PC = IABUS;                           ! Place Jump Address Into Program
                                      ! Counter
next;

/*************************************************************/
T = 2;                                ! Clock Cycle 2
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 2
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 2
      next;                           ! Execute Assignments

      /*******************************************************/
      T = 3;                          ! Clock Cycle 3
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];           ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];        ! On Data Bus And
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /*******************************************************/
      T = 2                           ! Return To Phase 2
                                      ! Of Clock Cycle 2
      );
      next;                           ! Execute Impending Assignments
```

```
/*****************************************************************/
T = 3;                                ! Clock Cycle 3
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 3
EXDBUF = DBUS;                        ! Instruction On Data Bus
                                      ! Is Placed In External Data
                                      ! Bus buffer
next;                                 ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                ! Clock Cycle 4
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 4
next;
PFR = EXDBUF;                         ! The Contents Of The External
                                      ! Data Bus Buffer Are Placed
                                      ! In Prefetch Register
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 4
ASN = hi;                             ! Deactivate Address Strobe
LDSN = hi;                            ! Deactivate Lower Data Strobe
UDSN = hi;                            ! Deactivate Upper Data Strobe
DTACKN = hi;                          ! Deactivate Data Transfer
                                      ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                ! Clock Cycle 5
next;                                 ! Execute Previous Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 5
RW = hi;                              ! Memory Read
ADENABLE = lo;                        ! Disable Address Bus Buffer
DBENABLE = lo;                        ! Disable Data Bus Buffer
IABUS = PC;                           ! Place PC On Internal Address
                                      ! Bus
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2 Of
PHI2 = hi;                            ! Clock Cycle 5
ADENABLE = hi;                        ! Enable Address Bus Buffer
FCMODE = SRMODE;                      ! User Mode
FCSPACE = 2;                          ! Accessing Program
EXABUF = IABUS;                       ! Gate Internal Address Bus
next;                                 ! Into External Address Buffer
ABUS = EXABUF;                        ! Address Placed On Bus(Added)
next;                                 ! Execute Pending Assignments
```

```
/****************************************************************/
T = 6;                                      ! Clock Cycle 6
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 6
ASN = lo;                                   ! Assert Address Strobe
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /**********************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /**********************************************************/
      T = 7                                ! Return To Phase 2
                                           ! Of Clock Cycle 7
      );
      next;                                ! Execute Impending Assignments

/****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
```

```
        PHI2 = hi;                          ! Of Clock Cycle 8
        EXDBUF = DBUS;                      ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
        next;                               ! Execute Pending Assignments

        /*************************************************************/
        T = 9;                              ! Clock Cycle 9
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1
        PHI2 = lo;                          ! Of Clock Cycle 9
        PFR = EXDBUF;                       ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
        next;                               ! Execute Pending Assignments

        PHI1 = lo;                          ! Phase 2
        PHI2 = hi;                          ! Of Clock Cycle 9
        ASN = hi;                           ! Deactivate Address Strobe
        LDSN = hi;                          ! Deactivate Lower Data Strobe
        UDSN = hi;                          ! Deactivate Upper Data Strobe
        PC = PC + 2;                        ! Increment Program Counter
        IR = PFR;                           ! Place Contents Of Prefetch
                                            ! Register Into Instruction
                                            ! Register
        DTACKN = hi;                        ! Deactivate Data Transfer
                                            ! Acknowledge(Added)
        next;                               ! Execute Pending Assignments
        T = 0                               ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                    (
                    case IR
                         0x3229,0x3429: move ! MOVE.W 4(A1),D1  [8(A1),D2]
                         0x027c: andi    ! AND.W #$DFFF,SR
                         047320: jmp    ! JMP (A0) If IR = Octal Value
                    esac
                    )

    main :=
         (
         power_on_initialize;
         fetch_initial_instruction;
         while READY eql hi
               (
               decode_execute_prefetch
               )
         )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W 04(A1,D7),D2 INSTRUCTION     */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                      Structure Declarations                           */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                  M68000 Programming Registers                         */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,             ! 8 Data Registers
A[0:6]<31:0>,             ! 7 Address Registers
UA7<31:0>,                ! User Stack Pointer
SA7<31:0>,                ! System Stack Pointer
PC<31:0>,                 ! Program Counter
SR<15:0>,                 ! Status Register

/*************************************************************************/
/*                                                                       */
/*                  Temporary Internal Registers                         */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                ! Prefetch Register
IR<15:0>,                 ! Instruction Register
FC<2:0>,                  ! Function Code Register
EXDBUF<15:0>,             ! External Data Bus Buffer Register
EXABUF<23:1>,             ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,            ! ALU Buffer 1
ALUBUF2<31:0>,            ! ALU Buffer 2
DTEMP<15:0>,              ! Temporary Data Storage
DISREG<31:0>,             ! Temporary Displacement Storage
SRTEMP<15:0>,             ! Temporary Status Register Storage
                          ! (Exception Processing)
IRTEMP<15:0>,             ! Temporary Instruction Register Storage
                          ! (Exception Processing)
TEMPADR<31:0>,            ! Temporary Cycle Address Storage
                          ! (Exception Processing)
ACTYPE<15:0>,             ! Temporary Access Type Storage
                          ! (Exception Processing)
VECADR<23:0>,             ! Temporary Vector Address Storage
                          ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/*********************************************************************/
/*                                                                   */
/*      Model transformation modifications:                          */
/*                                                                   */
/*           1) CDL decoder structure nonexistent in ISP' and un-    */
/*      necessary for model. Eliminated.                             */
/*           2) Multi-phase clock structure nonexistent in ISP'.     */
/*      Operations on registers will provide its equivalent.         */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                        */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                  */
/*           5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing   */
/*      MC68000 assembler and linker/loader models. The memory was   */
/*      also reduced from 8 Mwords to 32 Kbytes.                     */
/*                                                                   */
/*********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
twait<4:0>,                  ! Wait State Counter
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                   */
/*              External Address and Data Bus                        */
/*                                                                   */
/*********************************************************************/

DBUS<15:0>,                  ! External Data Bus
```

```
ABUS<23:1>;                     ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                        Register Subfields                          */
/*                                                                    */
/**********************************************************************/

PCADDR      = PC<23:0>,       ! Program Counter Address Field
SRTRACE     = SR<15>,         ! Trace Bit
SRMODE      = SR<13>,         ! Mode Selection Bit
SRCARRY     = SR<0>,          ! Carry Bit
SROVER      = SR<1>,          ! Overflow Bit
SRZERO      = SR<2>,          ! Zero Bit
SRNEG       = SR<3>,          ! Negative Bit
SREX        = SR<4>,          ! Extend Bit
SRMASK      = SR<10:8>,       ! Interrupt Mask
FCSPACE     = FC<1:0>,        ! Memory Access Address Space
FCMODE      = FC<2>,          ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,       ! PC Low Word
PCHI        = PC<31:16>,      ! PC High Word
D0LWORD     = D[0]<15:0>,     ! D[0] Low Word
D1LWORD     = D[1]<15:0>,     ! D[1] Low Word
D2LWORD     = D[2]<15:0>,     ! D[2] Low Word
D3LWORD     = D[3]<15:0>,     ! D[3] Low Word
D4LWORD     = D[4]<15:0>,     ! D[4] Low Word
D5LWORD     = D[5]<15:0>,     ! D[5] Low Word
D6LWORD     = D[6]<15:0>,     ! D[6] Low Word
D7LWORD     = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                   16K 16-Bit Word Internal Memory                  */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                       Logic Level Macros                           */
/*                                                                        */
/**************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/**************************************************************************/
/*                                                                        */
/*  Power On and Initialization. This process was not modeled but is     */
/*  added to initialize signals and registers.                            */
/*                                                                        */
/**************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                        ! Turn Power On
        next;                               ! Execute Assignment
        READY = lo;                         ! System Not Ready
        RESET = lo;                         ! Assert Reset For
        delay(100);                         ! 100 Miliseconds(Active Low)
        RESET = hi;                         ! Deactivate Reset
        next;                               ! Execute Pending Assignments
        ASN = hi;                           ! Initialize Address Strobe
        LDSN = hi;                          ! Initialize Lower Data Strobe
        UDSN = hi;                          ! Initialize Upper Data Strobe
        DTACKN = hi;                        ! Initialize Data Transfer Acknowledge
        RW = hi;                            ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                     ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                    ! Place Memory Locations Following The
        M[0x100f] = 0xff;                     ! JMP Instruction In A High State
        HALT = hi;                          ! Initialize Halt Flip-Flop(Active
                                            ! Low)
        T = 0;                              ! Initialize Clock Cycle Counter
        READY = hi;                         ! System Ready
        /******************************************************************/
        /*                                                                */
        /*     Routine Initialization Per Hamby and Guillory              */
        /*                                                                */
        /******************************************************************/
        D[7] = 0x00000006;              ! Place 6 Into D[7]
        A[0] = 0x1004;                      ! Place Hex 1004 Into A[0]
        A[1] = 0x2000;                  ! Store Data At This Address
        M[0x200a] = 0x55;                ! Data To Be Moved
        M[0x200b] = 0x55;
        PC = 0x1000;                        ! Place Hex 1000 Into Program Counter
        next                                ! Execute Assignments
        )

/**************************************************************************/
/*                                                                        */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary    */
/*  to retrieve modeled instructions for simulation and analysis. It    */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /****************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
```

```
PHI2 = lo;                                  ! Of Clock Cycle 2
while DTACKN eql hi                         ! Wait For Memory To Place
     (                                      ! Data On The Bus
     next;                                  ! Execute Impending Assignments

     PHI1 = lo;                             ! Phase 2
     PHI2 = hi;                             ! Of Clock Cycle 2
     next;                                  ! Execute Assignments

     /*******************************************************/
     T = 3;                                 ! Clock Cycle 3
     next;                                  ! Execute Assignment

     PHI1 = hi;                             ! Phase 1
     PHI2 = lo;                             ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
     DTACKN = lo;                           ! Asserts DTACKN(Added)
     next;                                  ! Execute Pending Assignments

     /*******************************************************/
     T = 2                                  ! Return To  Phase 2
                                            ! Of Clock Cycle 2
     );
     next;                                  ! Execute Impending Assignments

/***********************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/***********************************************************/
T = 4;                                      ! Clock Cycle 4
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 4
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 4
ASN = hi;                                   ! Deactivate Address Strobe
LDSN = hi;                                  ! Deactivate Lower Data Strobe
UDSN = hi;                                  ! Deactivate Upper Data Strobe
```

```
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 4;                            ! Increment Program Counter
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

  andi :=                                       ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                             ! Effect Of Instruction
       IR<15:8> = M[PC];                        ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                    ! Is To Set Status Register
          PC = PC + 2;                            ! Increment Program Counter
       T = 5;                                   ! Supervisor Bit To User
       next;                                    ! Mode
       T = 0                                    ! Requires 6 Clock Cycles
       )

  move :=                                       ! MOVE.W 4(A1,D7),D2 [D3]
        (

        /******************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        ABUS = 0xffffff;                        ! Address Bus High Impedanced
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        ABUS = IABUS<23:1>;                     ! Address Placed On Bus
        next;                                   ! Execute Impending Assignments

        /******************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
```

```
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 2
    next;                               ! Execute Assignments

    /***********************************************************/
    T = 3;                              ! Clock Cycle 3
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 3
    DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
    DTACKN = lo;                        ! Asserts DTACKN(Added)
    next;                               ! Execute Pending Assignments

    /***********************************************************/
    T = 2                               ! Return To  Phase 2
                                        ! Of Clock Cycle 2
    );
    next;                               ! Execute Impending Assignments

/****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
DISREG = EXDBUF<7:0> sxt 32;            ! Store Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
PC = PC + 2;                            ! Increment Program Counter
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
DISREG = DISREG + A[1];                 ! Add A[1] To Displacement Register
next;

/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBUS = 0xffff;                          ! Data Bus High Impedanced
DISREG = DISREG + D[7];               ! Add Data Register To Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
next;                                   ! Into External Address Buffer

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Previous Assignment
```

```
PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 7
RW = hi;                               ! Memory Read
ADENABLE = lo;                         ! Disable Address Bus Buffer
DBENABLE = lo;                         ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;                ! Place PC On Internal Address
                                       ! Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2 Of
PHI2 = hi;                             ! Clock Cycle 7
ADENABLE = hi;                         ! Enable Address Bus Buffer
FCMODE = SRMODE;                       ! User Mode
FCSPACE = 2;                           ! Accessing Data
EXABUF = IABUS<23:1>;                  ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                    ! Place Address On Bus
next;                                  ! Into External Address Buffer

/*************************************************************/
T = 8;                                 ! Clock Cycle 8
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 8
UDSN = lo;                             ! Activate Upper And
LDSN = lo;                             ! Lower Data Strobes
ASN = lo;                              ! Assert Address Strobe
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 8
next;                                  ! Execute Pending Assignments

/*************************************************************/
T = 9;                                 ! Clock Cycle 9
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 9
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 9
      next;                            ! Execute Assignments

      /*******************************************************/
      T = 10;                           ! Clock Cycle 10
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
```

```
          PHI2 = lo;                         ! Of Clock Cycle 10
          DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
          DTACKN = lo;                       ! Asserts DTACKN(Added)
          next;                              ! Execute Pending Assignments

          /*********************************************************/
          T = 9                              ! Return To  Phase 2
                                             ! Of Clock Cycle 9
          );
          next;                              ! Execute Impending Assignments

     /*************************************************************/
     T = 10;                                  ! Clock Cycle 10
     next;                                   ! Execute Assignment

     PHI1 = lo;                              ! Phase 2
     PHI2 = hi;                              ! Of Clock Cycle 10
     EXDBUF = DBUS;                          ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
     next;                                   ! Execute Pending Assignments

     /*************************************************************/
     T = 11;                                  ! Clock Cycle 11
     next;                                   ! Execute Assignment

     PHI1 = hi;                              ! Phase 1
     PHI2 = lo;                              ! Of Clock Cycle 11
     PFR = EXDBUF;                           ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
     next;                                   ! Execute Pending Assignments

     PHI1 = lo;                              ! Phase 2
     PHI2 = hi;                              ! Of Clock Cycle 11
     ASN = hi;                               ! Deactivate Address Strobe
     LDSN = hi;                              ! Deactivate Lower Data Strobe
     UDSN = hi;                              ! Deactivate Upper Data Strobe
     DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                             ! Acknowledge
     PC = PC + 2;                            ! Increment PC
     next;                                   ! Execute Pending Assignments

     /*************************************************************/
     T = 12;                                  ! Clock Cycle 12
     next;                                   ! Execute Assignment

     PHI1 = hi;                              ! Phase 1 Of
     PHI2 = lo;                              ! Clock Cycle 12

     RW = hi;                                ! Memory Read
     ADENABLE = lo;                          ! Disable Address Bus Buffer
```

```
ABUS = 0xffffff;                       ! Address Bus High Impedanced
DBUS = 0xffff;                         ! Data Bus Returned To High
                                       ! Impedance State
DBENABLE = lo;                         ! Disable Data Bus Buffer
IABUS = DISREG;                        ! Place DISREG On Internal Address
                                       ! Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2 Of
PHI2 = hi;                             ! Clock Cycle 12
ADENABLE = hi;                         ! Enable Address Bus Buffer
FCMODE = SRMODE;                       ! User Mode
FCSPACE = 1;                           ! Accessing Data
EXABUF = IABUS<23:1>;                  ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                    ! Place Address On Bus
SRCARRY = 0;                           ! Initialize Status Register
SROVER = 0;                            ! Condition Bits
SRZERO = 0;
SRNEG = 0;
next;                                  ! Into External Address Buffer

/*****************************************************************/

T = 13;                                 ! Clock Cycle 13
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 13
UDSN = lo;                             ! Activate Upper And
LDSN = lo;                             ! Lower Data Strobes
ASN = lo;                              ! Assert Address Strobe
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 13
next;                                  ! Execute Pending Assignments

/*****************************************************************/

T = 14;                                 ! Clock Cycle 14
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 14
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 14
      next;                            ! Execute Assignments
```

```
          /*************************************************************/
          T = 15;                                 ! Clock Cycle 15
          next;                                   ! Execute Assignment

          PHI1 = hi;                              ! Phase 1
          PHI2 = lo;                              ! Of Clock Cycle 15
          DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
          DTACKN = lo;                            ! Asserts DTACKN(Added)
          next;                                   ! Execute Pending Assignments

          /*************************************************************/
          T = 14                                  ! Return To  Phase 2
                                                  ! Of Clock Cycle 14
          );
          next;                                   ! Execute Impending Assignments

     /******************************************************************/
     T = 15;                                      ! Clock Cycle 15
     next;                                        ! Execute Assignment

     PHI1 = lo;                                   ! Phase 2
     PHI2 = hi;                                   ! Of Clock Cycle 15
     EXDBUF = DBUS;                               ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
     next;                                        ! Execute Pending Assignments

     /******************************************************************/
     T = 16;                                      ! Clock Cycle 16
     next;                                        ! Execute Assignment

     PHI1 = hi;                                   ! Phase 1
     PHI2 = lo;                                   ! Of Clock Cycle 16
     IDBUS = EXDBUF;
     if EXDBUF eql 0                              ! Set Condition Code Bits
        SRZERO = hi;                              ! As Appropriate
     if EXDBUF<15>
        SRNEG = hi;
     next;                                        ! Execute Pending Assignments

     PHI1 = lo;                                   ! Phase 2
     PHI2 = hi;                                   ! Of Clock Cycle 16
     if IR eql 0x3431                             ! Place Value In Either
        D[2] = IDBUS                              ! D[2] Or D[3]
     else                                         ! Depending On Instruction
        D[3] = IDBUS;
     ASN = hi;                                    ! Deactivate Address Strobe
     LDSN = hi;                                   ! Deactivate Lower Data Strobe
     UDSN = hi;                                   ! Deactivate Upper Data Strobe
     DTACKN = hi;                                 ! Deactivate Data Transfer(Added)
                                                  ! Acknowledge
     IR = PFR;
```

```
        next;                                       ! Execute Pending Assignments

        T = 0
        )

    jmp :=                                          ! JMP (A0)
        (

        /**************************************************************/

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 0
        DBUS = 0xffff;                              ! Place Data Bus In A High Impedance
        RW = hi;                                    ! Memory Read
        ADENABLE = lo;                              ! Disable Address Bus Buffer
        DBENABLE = lo;                              ! Disable Data Bus Buffer
        IABUS = PC;                                 ! Place PC On Internal Address
                                                    ! Bus
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2 Of
        PHI2 = hi;                                  ! Clock Cycle 0
        ADENABLE = hi;                              ! Enable Address Bus Buffer
        EXABUF = IABUS;                             ! Gate Internal Address Bus
                                                    ! Into External Address Buffer
        FCMODE = SRMODE;                            ! User Mode
        FCSPACE = 2;                                ! Accessing Program
        next;                                       ! Execute Pending Assignments
        ABUS = EXABUF;                              ! Address Placed On Bus(Added)
        next;                                       ! Execute Pending Assignments

        /**************************************************************/
        T = 1;                                      ! Clock Cycle 1
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 1
        ASN = lo;                                   ! Assert Address Strobe
        LDSN = lo;                                  ! Assert Lower Data Strobe
        UDSN = lo;                                  ! Assert Upper Data Strobe
        IABUS = A[0];                               ! Move Jump Address From A[0]
                                                    ! To Internal Address Buffer
        DBENABLE = hi;                              ! Enable Data Bus
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 1
        PC = IABUS;                                 ! Place Jump Address Into Program
                                                    ! Counter
        next;

        /**************************************************************/
        T = 2;                                      ! Clock Cycle 2
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 2
    next;                               ! Execute Assignments

    /*************************************************************/
    T = 3;                              ! Clock Cycle 3
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 3
    DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
    DTACKN = lo;                        ! Asserts DTACKN(Added)
    next;                               ! Execute Pending Assignments

    /*************************************************************/
    T = 2                               ! Return To Phase 2
                                        ! Of Clock Cycle 2
    );
    next;                               ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
```

```
PHI2 = hi;                          ! Of Clock Cycle 4
ASN = hi;                           ! Deactivate Address Strobe
LDSN = hi;                          ! Deactivate Lower Data Strobe
UDSN = hi;                          ! Deactivate Upper Data Strobe
DTACKN = hi;                        ! Deactivate Data Transfer
                                    ! Acknowledge(Added)
next;
/**************************************************************/
T = 5;                              ! Clock Cycle 5
next;                               ! Execute Previous Assignment

PHI1 = hi;                          ! Phase 1 Of
PHI2 = lo;                          ! Clock Cycle 5
RW = hi;                            ! Memory Read
ADENABLE = lo;                      ! Disable Address Bus Buffer
DBENABLE = lo;                      ! Disable Data Bus Buffer
IABUS = PC;                         ! Place PC On Internal Address
                                    ! Bus
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2 Of
PHI2 = hi;                          ! Clock Cycle 5
ADENABLE = hi;                      ! Enable Address Bus Buffer
FCMODE = SRMODE;                    ! User Mode
FCSPACE = 2;                        ! Accessing Program
EXABUF = IABUS;                     ! Gate Internal Address Bus
next;                               ! Into External Address Buffer
ABUS = EXABUF;                      ! Address Placed On Bus(Added)
next;                               ! Execute Pending Assignments

/**************************************************************/
T = 6;                              ! Clock Cycle 6
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1 Of
PHI2 = lo;                          ! Clock Cycle 6
ASN = lo;                           ! Assert Address Strobe
LDSN = lo;                          ! Assert Lower Data Strobe
UDSN = lo;                          ! Assert Upper Data Strobe
DBENABLE = hi;                      ! Enable Data Bus
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2
PHI2 = hi;                          ! Of Clock Cycle 6
next;                               ! Execute Pending Assignments

/**************************************************************/
T = 7;                              ! Clock Cycle 7
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1
PHI2 = lo;                          ! Of Clock Cycle 7
while DTACKN eql hi                 ! Wait For Memory To Place
```

```
        (                                        ! Data On The Bus
        next;                                    ! Execute Impending Assignments

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 7
        next;                                    ! Execute Assignments

        /************************************************************/
        T = 8;                                   ! Clock Cycle 8
        next;                                    ! Execute Assignment

        PHI1 = hi;                               ! Phase 1
        PHI2 = lo;                               ! Of Clock Cycle 8
        DBUS<15:8> = M[ABUS];                    ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];                 ! On Data Bus And
        DTACKN = lo;                             ! Asserts DTACKN(Added)
        next;                                    ! Execute Pending Assignments

        /************************************************************/
        T = 7                                   ! Return To Phase 2
                                                ! Of Clock Cycle 7
        );
        next;                                   ! Execute Impending Assignments

/*******************************************************************/
T = 8;                                           ! Clock Cycle 8
next;                                            ! Execute Assignment

PHI1 = lo;                                       ! Phase 2
PHI2 = hi;                                       ! Of Clock Cycle 8
EXDBUF = DBUS;                                   ! Instruction On Data Bus
                                                 ! Is Placed In External Data
                                                 ! Bus Buffer
next;                                            ! Execute Pending Assignments

/*******************************************************************/
T = 9;                                           ! Clock Cycle 9
next;                                            ! Execute Assignment

PHI1 = hi;                                       ! Phase 1
PHI2 = lo;                                       ! Of Clock Cycle 9
PFR = EXDBUF;                                    ! The Contents Of The External
                                                 ! Data Bus Buffer Are Placed
                                                 ! In Prefetch Register
next;                                            ! Execute Pending Assignments

PHI1 = lo;                                       ! Phase 2
PHI2 = hi;                                       ! Of Clock Cycle 9
ASN = hi;                                        ! Deactivate Address Strobe
LDSN = hi;                                       ! Deactivate Lower Data Strobe
UDSN = hi;                                       ! Deactivate Upper Data Strobe
PC = PC + 2;                                     ! Increment Program Counter
IR = PFR;                                        ! Place Contents Of Prefetch
```

```
                                                        ! Register Into Instruction
                                                        ! Register
        DTACKN = hi;                                    ! Deactivate Data Transfer
                                                        ! Acknowledge(Added)
        next;                                           ! Execute Pending Assignments
        T = 0                                           ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             0x3431,0x3631: move ! MOVE.W 4(A1,D7),D2 [D3]
                             0x027c: andi    ! AND.W #$DFFF,SR
                             047320: jmp    ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
     (
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           (
           decode_execute_prefetch
           )
     )
```

```
/***************************************************************************/
/*                                                                         */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W $2004,D5 INSTRUCTION          */
/*                                                                         */
/***************************************************************************/

/***************************************************************************/
/*                                                                         */
/*                      Structure Declarations                             */
/*                                                                         */
/***************************************************************************/

state

/***************************************************************************/
/*                                                                         */
/*                  M68000 Programming Registers                           */
/*                                                                         */
/***************************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/***************************************************************************/
/*                                                                         */
/*                  Temporary Internal Registers                           */
/*                                                                         */
/***************************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                     ! Reset Flip-Flop
HALT,                      ! Halt Flip-Flop
RW,                        ! Read/Write Flip-Flop
ADENABLE,                  ! Address Bus Buffer Enable
DBENABLE,                  ! Data Bus Buffer Enable
ASN,                       ! Address Strobe Flip-Flop
LDSN,                      ! Lower Data Strobe Flip-Flop
UDSN,                      ! Upper Data Strobe Flip-Flop
DTACKN,                    ! Data Transfer Acknowledge Flip-Flop
COUT,                      ! Carry Flip-Flop
EXCEPT,                    ! Exception Processing Flip-Flop
READY,                     ! Ready Flip-Flop

/*************************************************************************/
/*                                                                     */
/*      Model transformation modifications:                            */
/*                                                                     */
/*           1) CDL decoder structure nonexistent in ISP' and un-      */
/*      necessary for model. Eliminated.                               */
/*           2) Multi-phase clock structure nonexistent in ISP'.       */
/*      Operations on registers will provide its equivalent.           */
/*           3) Switch structure nonexistent in ISP'. Operation on a   */
/*      register will provide its equivalent.                          */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model   */
/*      equivalency and simplicity.                                    */
/*           5) The memory word length was reduced from 16 to 8 bit    */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing     */
/*      MC68000 assembler and linker/loader models. The memory was     */
/*      also reduced from 8 Mwords to 32 Kbytes.                       */
/*                                                                     */
/*************************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
twait<4:0>,                  ! Wait State Counter
SWITCH,                    ! Power Switch
PHI1,                      ! Phase 1 Of Two-Phase Clock
PHI2;                      ! Phase 2 Of Two-Phase Clock

port

/*************************************************************************/
/*                                                                     */
/*                External Address and Data Bus                       */
/*                                                                     */
/*************************************************************************/

DBUS<15:0>,                  ! External Data Bus
```

```
ABUS<23:1>;                  ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                       Register Subfields                           */
/*                                                                    */
/**********************************************************************/

PCADDR      = PC<23:0>,       ! Program Counter Address Field
SRTRACE     = SR<15>,         ! Trace Bit
SRMODE      = SR<13>,         ! Mode Selection Bit
SRCARRY     = SR<0>,          ! Carry Bit
SROVER      = SR<1>,          ! Overflow Bit
SRZERO      = SR<2>,          ! Zero Bit
SRNEG       = SR<3>,          ! Negative Bit
SREX        = SR<4>,          ! Extend Bit
SRMASK      = SR<10:8>,       ! Interrupt Mask
FCSPACE     = FC<1:0>,        ! Memory Access Address Space
FCMODE      = FC<2>,          ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,       ! PC Low Word
PCHI        = PC<31:16>,      ! PC High Word
D0LWORD     = D[0]<15:0>,     ! D[0] Low Word
D1LWORD     = D[1]<15:0>,     ! D[1] Low Word
D2LWORD     = D[2]<15:0>,     ! D[2] Low Word
D3LWORD     = D[3]<15:0>,     ! D[3] Low Word
D4LWORD     = D[4]<15:0>,     ! D[4] Low Word
D5LWORD     = D[5]<15:0>,     ! D[5] Low Word
D6LWORD     = D[6]<15:0>,     ! D[6] Low Word
D7LWORD     = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                   16K 16-Bit Word Internal Memory                  */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                       Logic Level Macros                           */
/*                                                                      */
/************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/************************************************************************/
/*                                                                      */
/*  Power On and Initialization. This process was not modeled but is  */
/*  added to initialize signals and registers.                         */
/*                                                                      */
/************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                      ! Turn Power On
        next;                             ! Execute Assignment
        READY = lo;                       ! System Not Ready
        RESET = lo;                       ! Assert Reset For
        delay(100);                       ! 100 Miliseconds(Active Low)
        RESET = hi;                       ! Deactivate Reset
        next;                             ! Execute Pending Assignments
        ASN = hi;                         ! Initialize Address Strobe
        LDSN = hi;                        ! Initialize Lower Data Strobe
        UDSN = hi;                        ! Initialize Upper Data Strobe
        DTACKN = hi;                      ! Initialize Data Transfer Acknowledge
        RW = hi;                          ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                   ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                 ! Place Memory Locations Following The
        M[0x100f] = 0xff;                  ! JMP Instruction In A High State
        HALT = hi;                        ! Initialize Halt Flip-Flop(Active
                                          ! Low)
        T = 0;                            ! Initialize Clock Cycle Counter
        READY = hi;                       ! System Ready
        /****************************************************************/
        /*                                                              */
        /*      Routine Initialization Per Hamby and Guillory          */
        /*                                                              */
        /****************************************************************/
        M[0x2004] = 0x55;                   ! Data To Be Moved
        M[0x2005] = 0x55;
        A[0] = 0x1004;                    ! Place Hex 1004 Into A[0]
        PC = 0x1000;                      ! Place Hex 1000 Into Program Counter
        next                              ! Execute Assignments
        )

/************************************************************************/
/*                                                                      */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was foshsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/**********************************************************************/

fetch_initial_instruction :=
      (

      /****************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
      while DTACKN eql hi                     ! Wait For Memory To Place
```

```
          (                                 ! Data On The Bus
          next;                             ! Execute Impending Assignments

          PHI1 = lo;                        ! Phase 2
          PHI2 = hi;                        ! Of Clock Cycle 2
          next;                             ! Execute Assignments

          /**********************************************************/
          T = 3;                            ! Clock Cycle 3
          next;                             ! Execute Assignment

          PHI1 = hi;                        ! Phase 1
          PHI2 = lo;                        ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
          DTACKN = lo;                      ! Asserts DTACKN(Added)
          next;                             ! Execute Pending Assignments

          /**********************************************************/
          T = 2                             ! Return To  Phase 2
                                            ! Of Clock Cycle 2
          );
          next;                             ! Execute Impending Assignments

      /**************************************************************/
      T = 3;                                ! Clock Cycle 3
      next;                                 ! Execute Assignment

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 3
      EXDBUF = DBUS;                        ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
      next;                                 ! Execute Pending Assignments

      /**************************************************************/
      T = 4;                                ! Clock Cycle 4
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 4
      PFR = EXDBUF;                         ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 4
      ASN = hi;                             ! Deactivate Address Strobe
      LDSN = hi;                            ! Deactivate Lower Data Strobe
      UDSN = hi;                            ! Deactivate Upper Data Strobe
      IR = PFR;                             ! Contents Of Prefetch Register
                                            ! Are Placed Into Instruction
```

```
                                        ! Register
     DTACKN = hi;                       ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;                       ! Increment Program Counter
     next;                              ! Execute Pending Assignments
     T = 0                              ! Reset Clock Cycle Counter
     )

andi :=                                 ! AND.W #$DFFF,SR
    (
    SRMODE = lo;                        ! Effect Of Instruction
    IR<15:8> = M[PC];                   ! Prefetch Next Instruction
    IR<7:0> = M[PC + 1];
    next;                               ! Is To Set Status Register
       PC = PC + 2;                       ! Increment Program Counter
    T = 5;                              ! Supervisor Bit To User
    next;                               ! Mode
    T = 0                               ! Requires 6 Clock Cycles
    )

move :=                                 ! MOVE.W $2004,D5 [D6]
     (

     /**************************************************************/

     PHI1 = hi;                         ! Phase 1 Of
     PHI2 = lo;                         ! Clock Cycle 0
     DBUS = 0xffff;                     ! Place Data Bus In High Impedance
     RW = hi;                           ! Memory Read
     ADENABLE = lo;                     ! Disable Address Bus Buffer
     ABUS = 0xffffff;                   ! Address Bus High Impedanced
     DBENABLE = lo;                     ! Disable Data Bus Buffer
     IABUS<31:1> = PC<31:1>;            ! Place PC On Internal Address
                                        ! Bus
     next;                              ! Execute Pending Assignments

     PHI1 = lo;                         ! Phase 2 Of
     PHI2 = hi;                         ! Clock Cycle 0
     ADENABLE = hi;                     ! Enable Address Bus Buffer
     EXABUF = IABUS<23:1>;              ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;                   ! User Mode
     FCSPACE = 2;                       ! Accessing Program
     ABUS = IABUS<23:1>;                ! Address Placed On Bus
     next;                              ! Execute Impending Assignments

     /**************************************************************/
     T = 1;                             ! Clock Cycle 1
     next;                              ! Execute Assignment

     PHI1 = hi;                         ! Phase 1 Of
     PHI2 = lo;                         ! Clock Cycle 1
     ASN = lo;                          ! Assert Address Strobe
```

```
LDSN = lo;                                  ! Assert Lower Data Strobe
UDSN = lo;                                  ! Assert Upper Data Strobe
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 1
next;                                       ! Execute Pending Assignments

/*************************************************************/
T = 2;                                      ! Clock Cycle 2
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 2
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 2
      next;                                 ! Execute Assignments

      /******************************************************/
      T = 3;                                ! Clock Cycle 3
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /******************************************************/
      T = 2                                 ! Return To  Phase 2
                                            ! Of Clock Cycle 2
      );
      next;                                 ! Execute Impending Assignments

/*************************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*************************************************************/
T = 4;                                      ! Clock Cycle 4
```

11/

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
DISREG = EXDBUF sxt 32;                 ! Store Displacement
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
PC = PC + 2;                            ! Increment Program Counter
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/**************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBUS = 0xffff;                          ! Data Bus Returned To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Data
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                     ! Place Address On Bus
next;                                   ! Into External Address Buffer

/**************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
UDSN = lo;                              ! Activate Upper And
LDSN = lo;                              ! Lower Data Strobes
ASN = lo;                               ! Assert Address Strobe
```

```
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /***********************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /***********************************************************/
      T = 7                                 ! Return To  Phase 2
                                            ! Of Clock Cycle 7
      );
      next;                                 ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 8
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                      ! Clock Cycle 9
next;                                       ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 9
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 9
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
PC = PC + 2;                            ! Increment PC
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 10;                                  ! Clock Cycle 10
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 10
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
ABUS = 0xffffff;                        ! Address Bus High Impedanced
DBUS = 0xffff;                          ! Data Bus Returned To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = DISREG;                           ! Place DISREG On Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 10
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Data
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
ABUS = IABUS<23:1>;                     ! Place Address On Bus
SRCARRY = 0;                            ! Initialize Status Register
SROVER = 0;                             ! Condition Bits
SRZERO = 0;
SRNEG = 0;
next;                                   ! Into External Address Buffer

/*****************************************************************/
T = 11;                                  ! Clock Cycle 11
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 11
UDSN = lo;                              ! Activate Upper And
```

```
LDSN = lo;                                ! Lower Data Strobes
ASN = lo;                                 ! Assert Address Strobe
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 11
next;                                     ! Execute Pending Assignments

/**************************************************************/
T = 12;                                    ! Clock Cycle 12
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 12
while DTACKN eql hi                       ! Wait For Memory To Place
    (                                     ! Data On The Bus
    next;                                 ! Execute Impending Assignments

    PHI1 = lo;                            ! Phase 2
    PHI2 = hi;                            ! Of Clock Cycle 12
    next;                                 ! Execute Assignments

    /*********************************************************/
    T = 13;                                ! Clock Cycle 13
    next;                                 ! Execute Assignment

    PHI1 = hi;                            ! Phase 1
    PHI2 = lo;                            ! Of Clock Cycle 13
    DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
    DTACKN = lo;                          ! Asserts DTACKN(Added)
    next;                                 ! Execute Pending Assignments

    /*********************************************************/
    T = 12                                 ! Return To  Phase 2
                                          ! Of Clock Cycle 12
    );
    next;                                 ! Execute Impending Assignments

/**************************************************************/
T = 13;                                    ! Clock Cycle 13
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 13
EXDBUF = DBUS;                            ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
next;                                     ! Execute Pending Assignments

/**************************************************************/
T = 14;                                    ! Clock Cycle 14
```

```
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 14
        IDBUS = EXDBUF;
        if EXDBUF eql 0                         ! Set Condition Code Bits
           SRZERO = hi;                         ! As Appropriate
        if EXDBUF<15>
           SRNEG = hi;
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 14
        if IR eql 0x3a38                        ! Place Value In Either
           D[5] = IDBUS                         ! D[5] Or D[6]
        else                                    ! Depending On Instruction
           D[6] = IDBUS;
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        IR = PFR;
        next;                                   ! Execute Pending Assignments

        T = 0
        )

jmp :=                                          ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 1;                               ! Clock Cycle 1
next;                                ! Execute Assignment

PHI1 = hi;                           ! Phase 1 Of
PHI2 = lo;                           ! Clock Cycle 1
ASN = lo;                            ! Assert Address Strobe
LDSN = lo;                           ! Assert Lower Data Strobe
UDSN = lo;                           ! Assert Upper Data Strobe
IABUS = A[0];                        ! Move Jump Address From A[0]
                                     ! To Internal Address Buffer
DBENABLE = hi;                       ! Enable Data Bus
next;                                ! Execute Pending Assignments

PHI1 = lo;                           ! Phase 2
PHI2 = hi;                           ! Of Clock Cycle 1
PC = IABUS;                          ! Place Jump Address Into Program
                                     ! Counter
next;

/*****************************************************************/
T = 2;                               ! Clock Cycle 2
next;                                ! Execute Assignment

PHI1 = hi;                           ! Phase 1
PHI2 = lo;                           ! Of Clock Cycle 2
while DTACKN eql hi                  ! Wait For Memory To Place
      (                              ! Data On The Bus
      next;                          ! Execute Impending Assignments

      PHI1 = lo;                     ! Phase 2
      PHI2 = hi;                     ! Of Clock Cycle 2
      next;                          ! Execute Assignments

      /***********************************************************/
      T = 3;                         ! Clock Cycle 3
      next;                          ! Execute Assignment

      PHI1 = hi;                     ! Phase 1
      PHI2 = lo;                     ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];          ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];       ! On Data Bus And
      DTACKN = lo;                   ! Asserts DTACKN(Added)
      next;                          ! Execute Pending Assignments

      /***********************************************************/
      T = 2                          ! Return To Phase 2
                                     ! Of Clock Cycle 2
      );
      next;                          ! Execute Impending Assignments

/*****************************************************************/
```

```
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments
```

```
/***************************************************************/
T = 6;                                ! Clock Cycle 6
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1 Of
PHI2 = lo;                            ! Clock Cycle 6
ASN = lo;                             ! Assert Address Strobe
LDSN = lo;                            ! Assert Lower Data Strobe
UDSN = lo;                            ! Assert Upper Data Strobe
DBENABLE = hi;                        ! Enable Data Bus
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 6
next;                                 ! Execute Pending Assignments

/***************************************************************/
T = 7;                                ! Clock Cycle 7
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 7
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 7
      next;                           ! Execute Assignments

      /*********************************************************/
      T = 8;                          ! Clock Cycle 8
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];           ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];        ! On Data Bus And
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /*********************************************************/
      T = 7                          ! Return To Phase 2
                                     ! Of Clock Cycle 7
      );
      next;                          ! Execute Impending Assignments

/***************************************************************/
T = 8;                                ! Clock Cycle 8
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 8
```

```
        EXDBUF = DBUS;                          ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
        next;                                   ! Execute Pending Assignments

        /**************************************************************/
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             0x3a38,0x3c38: move   ! MOVE.W $2004,D5 [D6]
                             0x027c: andi    ! AND.W #$DFFF,SR
                             047320: jmp     ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
      (
      power_on_initialize;
      fetch_initial_instruction;
      while READY eql hi
            (
            decode_execute_prefetch
            )
      )
```

```
/**********************************************************************/
/*                                                                    */
/*    MOTOROLA MC68000 MODEL OF THE MOVE.W $2004,$2008 INSTRUCTION    */
/*                                                                    */
/**********************************************************************/

/**********************************************************************/
/*                                                                    */
/*                       Structure Declarations                       */
/*                                                                    */
/**********************************************************************/

state

/**********************************************************************/
/*                                                                    */
/*                   M68000 Programming Registers                     */
/*                                                                    */
/**********************************************************************/

D[0:7]<31:0>,              ! 8 Data Registers
A[0:6]<31:0>,              ! 7 Address Registers
UA7<31:0>,                 ! User Stack Pointer
SA7<31:0>,                 ! System Stack Pointer
PC<31:0>,                  ! Program Counter
SR<15:0>,                  ! Status Register

/**********************************************************************/
/*                                                                    */
/*                   Temporary Internal Registers                     */
/*                                                                    */
/**********************************************************************/

PFR<15:0>,                 ! Prefetch Register
IR<15:0>,                  ! Instruction Register
FC<2:0>,                   ! Function Code Register
EXDBUF<15:0>,              ! External Data Bus Buffer Register
EXABUF<23:1>,              ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,             ! ALU Buffer 1
ALUBUF2<31:0>,             ! ALU Buffer 2
DTEMP<15:0>,               ! Temporary Data Storage
DISREG<31:0>,              ! Temporary Displacement Storage
SRTEMP<15:0>,              ! Temporary Status Register Storage
                           ! (Exception Processing)
IRTEMP<15:0>,              ! Temporary Instruction Register Storage
                           ! (Exception Processing)
TEMPADR<31:0>,             ! Temporary Cycle Address Storage
                           ! (Exception Processing)
ACTYPE<15:0>,              ! Temporary Access Type Storage
                           ! (Exception Processing)
VECADR<23:0>,              ! Temporary Vector Address Storage
                           ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/*********************************************************************/
/*                                                                 */
/*      Model transformation modifications:                        */
/*                                                                 */
/*           1) CDL decoder structure nonexistent in ISP' and un-  */
/*      necessary for model. Eliminated.                           */
/*           2) Multi-phase clock structure nonexistent in ISP'.   */
/*      Operations on registers will provide its equivalent.       */
/*           3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                      */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                */
/*           5) The memory word length was reduced from 16 to 8 bit */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing */
/*      MC68000 assembler and linker/loader models. The memory was */
/*      also reduced from 8 Mwords to 32 Kbytes.                   */
/*                                                                 */
/*********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
twait<7:0>,                  ! Wait Cycle Counter
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                 */
/*              External Address and Data Bus                      */
/*                                                                 */
/*********************************************************************/

DBUS<15:0>,                  ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/*********************************************************************/
/*                                                                 */
/*                      Register Subfields                         */
/*                                                                 */
/*********************************************************************/

PCADDR        = PC<23:0>,      ! Program Counter Address Field
SRTRACE       = SR<15>,        ! Trace Bit
SRMODE        = SR<13>,        ! Mode Selection Bit
SRCARRY       = SR<0>,         ! Carry Bit
SROVER        = SR<1>,         ! Overflow Bit
SRZERO        = SR<2>,         ! Zero Bit
SRNEG         = SR<3>,         ! Negative Bit
SREX          = SR<4>,         ! Extend Bit
SRMASK        = SR<10:8>,      ! Interrupt Mask
FCSPACE       = FC<1:0>,       ! Memory Access Address Space
FCMODE        = FC<2>,         ! User/Supervisor Mode Bit
PCLOW         = PC<15:0>,      ! PC Low Word
PCHI          = PC<31:16>,     ! PC High Word
D0LWORD       = D[0]<15:0>,    ! D[0] Low Word
D1LWORD       = D[1]<15:0>,    ! D[1] Low Word
D2LWORD       = D[2]<15:0>,    ! D[2] Low Word
D3LWORD       = D[3]<15:0>,    ! D[3] Low Word
D4LWORD       = D[4]<15:0>,    ! D[4] Low Word
D5LWORD       = D[5]<15:0>,    ! D[5] Low Word
D6LWORD       = D[6]<15:0>,    ! D[6] Low Word
D7LWORD       = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW     = HANADR<15:0>, ! HANADR Low Word
HANADRHI      = HANADR<31:16>,! HANADR High Word
TEMPADRLOW    = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI     = TEMPADR<31:16>;! TEMPADR High Word

memory

/*********************************************************************/
/*                                                                 */
/*                   16K 16-Bit Word Internal Memory               */
/*                                                                 */
/*********************************************************************/

M[0:32767]<7:0>;

macro

/*********************************************************************/
/*                                                                 */
/*                      Logic Level Macros                         */
/*                                                                         */
/***************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/***************************************************************************/
/*                                                                         */
/*  Power On and Initialization. This process was not modeled but is      */
/*  added to initialize signals and registers.                             */
/*                                                                         */
/***************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        M[0x1010] = 0xff;              ! Place Memory Locations Following The
        M[0x1011] = 0xff;                ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /***************************************************************/
        /*                                                             */
        /*     Routine Initialization Per Hamby and Guillory           */
        /*                                                             */
        /***************************************************************/
        M[0x2004] = 0x55;               ! Data To Be Moved
        M[0x2005] = 0x55;
        A[0] = 0x1004;                  ! Place Hex 1004 Into A[0]
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/***************************************************************************/
/*                                                                         */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary       */
/*  to retrieve modeled instructions for simulation and analysis. It       */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                     */
/*                                                                     */
/*********************************************************************/

fetch_initial_instruction :=
      (

      /**************************************************************/

      PHI1 = hi;                           ! Phase 1 Of
      PHI2 = lo;                           ! Clock Cycle 0
      RW = hi;                             ! Memory Read
      ADENABLE = lo;                       ! Disable Address Bus Buffer
      DBENABLE = lo;                       ! Disable Data Bus Buffer
      IABUS = PC;                          ! Place PC On Internal Address
                                           ! Bus
      next;                                ! Execute Pending Assignments

      PHI1 = lo;                           ! Phase 2 Of
      PHI2 = hi;                           ! Clock Cycle 0
      ADENABLE = hi;                       ! Enable Address Bus Buffer
      EXABUF = IABUS;                      ! Gate Internal Address Bus
                                           ! Into External Address Buffer
      FCMODE = SRMODE;                     ! User Mode
      FCSPACE = 2;                         ! Accessing Program
      next;                                ! Execute Impending Assignments
      ABUS = EXABUF;                       ! Address Placed On Bus(Added)
      next;                                ! Execute Pending Assignments

      /**************************************************************/
      T = 1;                               ! Clock Cycle 1
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1 Of
      PHI2 = lo;                           ! Clock Cycle 1
      ASN = lo;                            ! Assert Address Strobe
      LDSN = lo;                           ! Assert Lower Data Strobe
      UDSN = lo;                           ! Assert Upper Data Strobe
      DBENABLE = hi;                       ! Enable Data Bus
      next;                                ! Execute Pending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 1
      next;                                ! Execute Pending Assignments

      /**************************************************************/
      T = 2;                               ! Clock Cycle 2
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 2
      while DTACKN eql hi                  ! Wait For Memory To Place
```

```
          (                                    ! Data On The Bus
          next;                                ! Execute Impending Assignments

          PHI1 = lo;                           ! Phase 2
          PHI2 = hi;                           ! Of Clock Cycle 2
          next;                                ! Execute Assignments

          /******************************************************/
          T = 3;                               ! Clock Cycle 3
          next;                                ! Execute Assignment

          PHI1 = hi;                           ! Phase 1
          PHI2 = lo;                           ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
          DTACKN = lo;                         ! Asserts DTACKN(Added)
          next;                                ! Execute Pending Assignments

          /******************************************************/
          T = 2                                ! Return To  Phase 2
                                               ! Of Clock Cycle 2
          );
          next;                                ! Execute Impending Assignments

      /**********************************************************/
      T = 3;                                   ! Clock Cycle 3
      next;                                    ! Execute Assignment

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 3
      EXDBUF = DBUS;                           ! Instruction On Data Bus
                                               ! Is Placed In External Data
                                               ! Bus Buffer
      next;                                    ! Execute Pending Assignments

      /**********************************************************/
      T = 4;                                   ! Clock Cycle 4
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 4
      PFR = EXDBUF;                            ! The Contents Of The External
                                               ! Data Bus Buffer Are Placed
                                               ! In Prefetch Register
      next;                                    ! Execute Pending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 4
      ASN = hi;                                ! Deactivate Address Strobe
      LDSN = hi;                               ! Deactivate Lower Data Strobe
      UDSN = hi;                               ! Deactivate Upper Data Strobe
      IR = PFR;                                ! Contents Of Prefetch Register
                                               ! Are Placed Into Instruction
```

```
                                             ! Register
        DTACKN = hi;                         ! Deactivate Data Transfer(Added)
                                             ! Acknowledge
        PC = PC + 4;                         ! Increment Program Counter
        next;                                ! Execute Pending Assignments
        T = 0                                ! Reset Clock Cycle Counter
        )

move :=                                      ! MOVE.W $2004,$2008
        (

        /*********************************************************************/

        PHI1 = hi;                           ! Phase 1 Of
        PHI2 = lo;                           ! Clock Cycle 0
        DBUS = 0xffff;                       ! Place Data Bus In High Impedance
        RW = hi;                             ! Memory Read
        ADENABLE = lo;                       ! Disable Address Bus Buffer
        DBENABLE = lo;                       ! Disable Data Bus Buffer
        IABUS = PC;                          ! Place PC On Internal Address
                                             ! Bus
        next;                                ! Execute Pending Assignments

        PHI1 = lo;                           ! Phase 2 Of
        PHI2 = hi;                           ! Clock Cycle 0
        ADENABLE = hi;                       ! Enable Address Bus Buffer
        EXABUF = IABUS;                      ! Gate Internal Address Bus
                                             ! Into External Address Buffer
        FCMODE = SRMODE;                     ! User Mode
        FCSPACE = 2;                         ! Accessing Program
        ABUS = IABUS;                        ! Address Placed On Bus(Added)
        next;                                ! Execute Pending Assignments

        /*********************************************************************/
        T = 1;                               ! Clock Cycle 1
        next;                                ! Execute Assignment

        PHI1 = hi;                           ! Phase 1 Of
        PHI2 = lo;                           ! Clock Cycle 1
        ASN = lo;                            ! Assert Address Strobe
        LDSN = lo;                           ! Assert Lower Data Strobe
        UDSN = lo;                           ! Assert Upper Data Strobe
        DBENABLE = hi;                       ! Enable Data Bus
        next;                                ! Execute Pending Assignments

        PHI1 = lo;                           ! Phase 2
        PHI2 = hi;                           ! Of Clock Cycle 1
        next;                                ! Execute Pending Assignments

        /*********************************************************************/
        T = 2;                               ! Clock Cycle 2
        next;                                ! Execute Assignment
```

```
PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 2
      next;                              ! Execute Assignments

      /*******************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*******************************************************/
      T = 2                              ! Return To  Phase 2
                                         ! Of Clock Cycle 2
      );
      next;                              ! Execute Impending Assignments

/***************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/***************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
TEMPADRHI = EXDBUF;                      ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Temporary Address Register
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 4
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
```

```
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
PC = PC + 2;                            ! Increment PC
next;

/******************************************************************/

T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
DBUS = 0xffff;                          ! Place Data Bus In High Impedance
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
ABUS = IABUS;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 7
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 7
      next;                           ! Execute Assignments

      /**************************************************************/
      T = 8;                          ! Clock Cycle 8
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];           ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];        ! On Data Bus And
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /**************************************************************/
      T = 7                           ! Return To  Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments

/********************************************************************/
T = 8;                                ! Clock Cycle 8
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 8
EXDBUF = DBUS;                        ! Instruction On Data Bus
                                      ! Is Placed In External Data
                                      ! Bus Buffer
next;                                 ! Execute Pending Assignments

/********************************************************************/
T = 9;                                ! Clock Cycle 9
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 9
TEMPADBRLOW = EXDBUF;                 ! The Contents Of The External
                                      ! Data Bus Buffer Are Placed
                                      ! In Temporary Register
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 9
ASN = hi;                             ! Deactivate Address Strobe
LDSN = hi;                            ! Deactivate Lower Data Strobe
```

```
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
PC = PC + 2;                            ! Increment PC
next;

/******************************************************************/

T = 10;                                  ! Clock Cycle 10
next;

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 10
DBUS = 0xffff;                          ! Place Data Bus In High Impedance
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 10
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
ABUS = IABUS;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 11;                                  ! Clock Cycle 11
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 11
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 11
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 12;                                  ! Clock Cycle 12
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 12
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 12
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 13;                            ! Clock Cycle 13
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 13
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 12                             ! Return To  Phase 2
                                        ! Of Clock Cycle 12
      );
      next;                             ! Execute Impending Assignments

/*************************************************************/
T = 13;                                  ! Clock Cycle 13
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 13
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 14;                                  ! Clock Cycle 14
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 14
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 14
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Are Placed Into Instruction
                                        ! Register
```

```
DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
PC = PC + 2;                              ! Increment PC
next;

/*****************************************************************/

T = 15;                                   ! Clock Cycle 15
next;

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 15
DBUS = 0xffff;                            ! Place Data Bus In High Impedance
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS = TEMPADR;                          ! Place TEMPADR On Internal Address
                                          ! Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 15
ADENABLE = hi;                            ! Enable Address Bus Buffer
EXABUF = IABUS;                           ! Gate Internal Address Bus
                                          ! Into External Address Buffer
FCMODE = SRMODE;                          ! User Mode
FCSPACE = 1;                              ! Accessing Program
TEMPADRHI = EXDBUF;                       ! Store High Word Of Destination
ABUS = IABUS;                            ! Address Placed On Bus(Added)
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 16;                                    ! Clock Cycle 16
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 16
ASN = lo;                                 ! Assert Address Strobe
LDSN = lo;                                ! Assert Lower Data Strobe
UDSN = lo;                                ! Assert Upper Data Strobe
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 16
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 17;                                    ! Clock Cycle 17
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 17
```

```
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 17
      next;                              ! Execute Assignments

      /*******************************************************/
      T = 18;                             ! Clock Cycle 18
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 18
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*******************************************************/
      T = 17                              ! Return To  Phase 2
                                         ! Of Clock Cycle 17
      );
      next;                              ! Execute Impending Assignments

/*************************************************************/
T = 18;                                   ! Clock Cycle 18
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 18
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/*************************************************************/
T = 19;                                   ! Clock Cycle 19
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 19
DTEMP = EXDBUF;                          ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Temporary Register
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 19
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
UDSN = hi;                               ! Deactivate Upper Data Strobe
                                         ! Are Placed Into Instruction
```

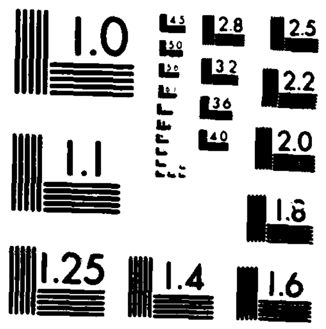
```
                                           ! Register
DTACKN = hi;                               ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
next;

/*****************************************************************/

T = 20;                                    ! Clock Cycle 20
next;

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 20
DBUS = 0xffff;                             ! Place Data Bus In High Impedance
RW = hi;                                   ! Memory Read
ADENABLE = lo;                             ! Disable Address Bus Buffer
DBENABLE = lo;                             ! Disable Data Bus Buffer
IABUS = PC;                                ! Place PC On Internal Address
                                           ! Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2 Of
PHI2 = hi;                                 ! Clock Cycle 20
ADENABLE = hi:                             ! Enable Address Bus Buffer
EXABUF = IABUS;                            ! Gate Internal Address Bus
                                           ! Into External Address Buffer
FCMODE = SRMODE;                           ! User Mode
FCSPACE = 2;                               ! Accessing Program
ABUS = IABUS;                             ! Address Placed On Bus(Added)
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 21;                                     ! Clock Cycle 21
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 21
ASN = lo;                                  ! Assert Address Strobe
LDSN = lo;                                 ! Assert Lower Data Strobe
UDSN = lo;                                 ! Assert Upper Data Strobe
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 21
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 22;                                     ! Clock Cycle 22
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 22
while DTACKN eql hi                        ! Wait For Memory To Place
```

```
        (                                        ! Data On The Bus
        next;                                    ! Execute Impending Assignments

        PHI1 = lo;                               ! Phase 2
        PHI2 = hi;                               ! Of Clock Cycle 22
        next;                                    ! Execute Assignments

        /*******************************************************/
        T = 23;                                   ! Clock Cycle 23
        next;                                    ! Execute Assignment

        PHI1 = hi;                               ! Phase 1
        PHI2 = lo;                               ! Of Clock Cycle 23
        DBUS<15:8> = M[ABUS];                    ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];                 ! On Data Bus And
        DTACKN = lo;                             ! Asserts DTACKN(Added)
        next;                                    ! Execute Pending Assignments

        /*******************************************************/
        T = 22                                    ! Return To  Phase 2
                                                 ! Of Clock Cycle 22
        );
        next;                                    ! Execute Impending Assignments

    /***********************************************************/
    T = 23;                                       ! Clock Cycle 23
    next;                                        ! Execute Assignment

    PHI1 = lo;                                   ! Phase 2
    PHI2 = hi;                                   ! Of Clock Cycle 23
    EXDBUF = DBUS;                               ! Instruction On Data Bus
                                                 ! Is Placed In External Data
                                                 ! Bus Buffer
    next;                                        ! Execute Pending Assignments

    /***********************************************************/
    T = 24;                                       ! Clock Cycle 24
    next;                                        ! Execute Assignment

    PHI1 = hi;                                   ! Phase 1
    PHI2 = lo;                                   ! Of Clock Cycle 24
    TEMPADRLOW = EXDBUF;                         ! The Contents Of The External
                                                 ! Data Bus Buffer Are Placed
                                                 ! In Temporary Register
    next;                                        ! Execute Pending Assignments

    PHI1 = lo;                                   ! Phase 2
    PHI2 = hi;                                   ! Of Clock Cycle 24
    ASN = hi;                                    ! Deactivate Address Strobe
    LDSN = hi;                                   ! Deactivate Lower Data Strobe
    UDSN = hi;                                   ! Deactivate Upper Data Strobe
                                                 ! Are Placed Into Instruction
                                                 ! Register.
```

```
DTACKN = hi;                               ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
PC = PC + 2;                               ! Increment PC
next;

/*****************************************************************/
T = 25;                                     ! Clock Cycle 25
next;                                      ! Execute Previous Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 25
RW = hi;                                   ! Memory Read
ADENABLE = lo;                             ! Disable Address Bus Buffer
DBUS = 0xffff;                             ! Data Bus Returned To High
                                           ! Impedance State
DBENABLE = lo;                             ! Disable Data Bus Buffer
IABUS = TEMPADR;                         ! Place TEMPADR On Internal Address
                                           ! Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2 Of
PHI2 = hi;                                 ! Clock Cycle 25
ADENABLE = hi;                             ! Enable Address Bus Buffer
FCMODE = SRMODE;                           ! User Mode
FCSPACE = 1;                               ! Accessing Program
EXABUF = IABUS;                            ! Gate Internal Address Bus
IDBUS = DTEMP;                             ! Place Low Word from DTEMP On
                                           ! Internal Data Bus
ABUS = IABUS;                             ! Address Placed On Bus(Added)
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 26;                                     ! Clock Cycle 26
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 26
ASN = lo;                                  ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                            ! Place Contents Of Internal
                                           ! Data Bus Into External Data Buffer
SRCARRY = lo;                              ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 26
if EXDBUF eql 0                            ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                             ! Place Data On External Data Bus
DBENABLE = hi;                             ! Enable Data Bus
```

AD-A164 257 THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. C A BAXLEY 4/5

UNCLASSIFIED DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
next;                                          ! Execute Pending Assignments

/******************************************************************/
T = 27;                                         ! Clock Cycle 27
next;                                          ! Execute Assignment

PHI1 = hi;                                     ! Phase 1
PHI2 = lo;                                     ! Of Clock Cycle 27
if EXDBUF<15>                                  ! Set Negative Condition Bit
   SRNEG = hi;                                 ! If Needed
UDSN = lo;                                     ! Activate Upper And
LDSN = lo;                                     ! Lower Data Strobes
twait = 0;                                     ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                            ! Wait For Memory To Place
      (                                        ! Data On The Bus
      twait = twait + 1;                       ! Increment Wait Cycle
      next;                                    ! Execute Impending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 27
      next;                                    ! Execute Assignments

      /***********************************************************/
      T = 28;                                   ! Clock Cycle 28
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 28
      if twait eql 2                           ! Memory Responds After 2 Cycles
      (
      M[ABUS] = DBUS<15:8>;                    ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;                 ! In Memory
      DTACKN = lo                            ! Asserts DTACKN(Added)
      );
      next;                                    ! Execute Pending Assignments

      /***********************************************************/
      T = 27                                   ! Return To Phase 2
                                             ! Of Clock Cycle 27
      );
      next;                                  ! Execute Impending Assignments

/******************************************************************/
T = 28;                                         ! Clock Cycle 28
next;                                          ! Execute Assignment

PHI1 = lo;                                     ! Phase 2
PHI2 = hi;                                     ! Of Clock Cycle 28
next;                                          ! Execute Pending Assignments

/******************************************************************/
T = 29;                                         ! Clock Cycle 29
```

```
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 29
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 29
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
UDSN = hi;                                ! Deactivate Upper Data Strobe
DTACKN = hi;                              ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
next;                                     ! Execute Pending Assignments

/***************************************************************/

T = 30;                                   ! Clock Cycle 30
next;

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 30
DBUS = 0xffff;                            ! Place Data Bus In High Impedance
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS = PC;                               ! Place PC On Internal Address
                                          ! Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 30
ADENABLE = hi;                            ! Enable Address Bus Buffer
EXABUF = IABUS;                           ! Gate Internal Address Bus
                                          ! Into External Address Buffer
FCMODE = SRMODE;                          ! User Mode
FCSPACE = 2;                              ! Accessing Program
ABUS = IABUS;                           ! Address Placed On Bus(Added)
next;                                     ! Execute Pending Assignments

/***************************************************************/
T = 31;                                     ! Clock Cycle 31
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 31
ASN = lo;                                 ! Assert Address Strobe
LDSN = lo;                                ! Assert Lower Data Strobe
UDSN = lo;                                ! Assert Upper Data Strobe
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
```

```
PHI2 = hi;                              ! Of Clock Cycle 31
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 32;                                  ! Clock Cycle 32
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 32
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 32
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 33;                            ! Clock Cycle 33
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 33
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 32                             ! Return To  Phase 2
                                        ! Of Clock Cycle 32
      );
      next;                             ! Execute Impending Assignments

/****************************************************************/
T = 33;                                  ! Clock Cycle 33
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 33
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 34;                                  ! Clock Cycle 34
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 34
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
```

```
                                                  ! In Prefetch Register
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 34
        ASN = hi;                                 ! Deactivate Address Strobe
        LDSN = hi;                                ! Deactivate Lower Data Strobe
        UDSN = hi;                                ! Deactivate Upper Data Strobe
                                                  ! Are Placed Into Instruction
                                                  ! Register
        DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                                  ! Acknowledge
        IR = PFR;                                 ! Load Instruction Register
                                                  ! With Prefetch Register
        PC = PC + 2;                              ! Increment PC
        next;

        T = 0
        )

  andi :=                                         ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                               ! Effect Of Instruction
       IR<15:8> = M[PC];                          ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                      ! Is To Set Status Register
          PC = PC + 2;                              ! Increment Program Counter
       T = 5;                                     ! Supervisor Bit To User
       next;                                      ! Mode
       T = 0                                      ! Requires 6 Clock Cycles
       )

  jmp :=                                          ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                                ! Phase 1 Of
        PHI2 = lo;                                ! Clock Cycle 0
        DBUS = 0xffff;                            ! Place Data Bus In A High Impedance
        RW = hi;                                  ! Memory Read
        ADENABLE = lo;                            ! Disable Address Bus Buffer
        DBENABLE = lo;                            ! Disable Data Bus Buffer
        IABUS = PC;                               ! Place PC On Internal Address
                                                  ! Bus
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2 Of
        PHI2 = hi;                                ! Clock Cycle 0
        ADENABLE = hi;                            ! Enable Address Bus Buffer
        EXABUF = IABUS;                           ! Gate Internal Address Bus
                                                  ! Into External Address Buffer
        FCMODE = SRMODE;                          ! User Mode
```

```
FCSPACE = 2;                               ! Accessing Program
next;                                      ! Execute Pending Assignments
ABUS = EXABUF;                             ! Address Placed On Bus(Added)
next;                                      ! Execute Pending Assignments

/****************************************************************/
T = 1;                                     ! Clock Cycle 1
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 1
ASN = lo;                                  ! Assert Address Strobe
LDSN = lo;                                 ! Assert Lower Data Strobe
UDSN = lo;                                 ! Assert Upper Data Strobe
IABUS = A[0];                              ! Move Jump Address From A[0]
                                           ! To Internal Address Buffer
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 1
PC = IABUS;                                ! Place Jump Address Into Program
                                           ! Counter
next;

/****************************************************************/
T = 2;                                     ! Clock Cycle 2
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 2
while DTACKN eql hi                        ! Wait For Memory To Place
      (                                    ! Data On The Bus
      next;                                ! Execute Impending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 2
      next;                                ! Execute Assignments

      /**********************************************************/
      T = 3;                               ! Clock Cycle 3
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
      DTACKN = lo;                         ! Asserts DTACKN(Added)
      next;                                ! Execute Pending Assignments

      /**********************************************************/
      T = 2                                ! Return To Phase 2
                                           ! Of Clock Cycle 2
```

```
        );
        next;                                 ! Execute Impending Assignments

    /***************************************************************/
    T = 3;                                    ! Clock Cycle 3
    next;                                     ! Execute Assignment

    PHI1 = lo;                                ! Phase 2
    PHI2 = hi;                                ! Of Clock Cycle 3
    EXDBUF = DBUS;                            ! Instruction On Data Bus
                                              ! Is Placed In External Data
                                              ! Bus Buffer
    next;                                     ! Execute Pending Assignments

    /***************************************************************/
    T = 4;                                    ! Clock Cycle 4
    next;                                     ! Execute Assignment

    PHI1 = hi;                                ! Phase 1
    PHI2 = lo;                                ! Of Clock Cycle 4
    next;
    PFR = EXDBUF;                             ! The Contents Of The External
                                              ! Data Bus Buffer Are Placed
                                              ! In Prefetch Register
    next;                                     ! Execute Pending Assignments

    PHI1 = lo;                                ! Phase 2
    PHI2 = hi;                                ! Of Clock Cycle 4
    ASN = hi;                                 ! Deactivate Address Strobe
    LDSN = hi;                                ! Deactivate Lower Data Strobe
    UDSN = hi;                                ! Deactivate Upper Data Strobe
    DTACKN = hi;                              ! Deactivate Data Transfer
                                              ! Acknowledge(Added)
    next;
    /***************************************************************/
    T = 5;                                    ! Clock Cycle 5
    next;                                     ! Execute Previous Assignment

    PHI1 = hi;                                ! Phase 1 Of
    PHI2 = lo;                                ! Clock Cycle 5
    RW = hi;                                  ! Memory Read
    ADENABLE = lo;                            ! Disable Address Bus Buffer
    DBENABLE = lo;                            ! Disable Data Bus Buffer
    IABUS = PC;                               ! Place PC On Internal Address
                                              ! Bus
    next;                                     ! Execute Pending Assignments

    PHI1 = lo;                                ! Phase 2 Of
    PHI2 = hi;                                ! Clock Cycle 5
    ADENABLE = hi;                            ! Enable Address Bus Buffer
    FCMODE = SRMODE;                          ! User Mode
    FCSPACE = 2;                              ! Accessing Program
    EXABUF = IABUS;                           ! Gate Internal Address Bus
```

```
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 7
      next;                             ! Execute Assignments

      /************************************************************/
      T = 8;                            ! Clock Cycle 8
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /************************************************************/
      T = 7                           ! Return To Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments

/******************************************************************/
T = 8;                                  ! Clock Cycle 8
```

```
        next;                                     ! Execute Assignment

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 8
        EXDBUF = DBUS;                            ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
        next;                                     ! Execute Pending Assignments

        /****************************************************************/
        T = 9;                                    ! Clock Cycle 9
        next;                                     ! Execute Assignment

        PHI1 = hi;                                ! Phase 1
        PHI2 = lo;                                ! Of Clock Cycle 9
        PFR = EXDBUF;                             ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Prefetch Register
        next;                                     ! Execute Pending Assignments

        PHI1 = lo;                                ! Phase 2
        PHI2 = hi;                                ! Of Clock Cycle 9
        ASN = hi;                                 ! Deactivate Address Strobe
        LDSN = hi;                                ! Deactivate Lower Data Strobe
        UDSN = hi;                                ! Deactivate Upper Data Strobe
        PC = PC + 2;                              ! Increment Program Counter
        IR = PFR;                                 ! Place Contents Of Prefetch
                                                  ! Register Into Instruction
                                                  ! Register
        DTACKN = hi;                              ! Deactivate Data Transfer
                                                  ! Acknowledge(Added)
        next;                                     ! Execute Pending Assignments
        T = 0                                     ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                            (
                            case IR
                                0x33f9: move      ! MOVE.W $2004,$2008
                                047320: jmp       ! JMP (A0) If IR = Octal Value
                                0x027c: andi      ! AND.W #$DFFF,SR
                            esac
                            )

    main :=
         (
         power_on_initialize;
         fetch_initial_instruction;
         while READY eql hi
               (
               decode_execute_prefetch
               )
         )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE MOVE.W #$5555,D1 INSTRUCTION        */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                     Structure Declarations                            */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                M68000 Programming Registers                           */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/*************************************************************************/
/*                                                                       */
/*                Temporary Internal Registers                           */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,              ! Temporary Address Storage For
                           ! Exception Handler Routine
T<7:0>,                    ! Clock Cycle Counter
RESET,                  ! Reset Flip-Flop
HALT,                   ! Halt Flip-Flop
RW,                     ! Read/Write Flip-Flop
ADENABLE,               ! Address Bus Buffer Enable
DBENABLE,               ! Data Bus Buffer Enable
ASN,                    ! Address Strobe Flip-Flop
LDSN,                   ! Lower Data Strobe Flip-Flop
UDSN,                   ! Upper Data Strobe Flip-Flop
DTACKN,                 ! Data Transfer Acknowledge Flip-Flop
COUT,                   ! Carry Flip-Flop
EXCEPT,                 ! Exception Processing Flip-Flop
READY,                  ! Ready Flip-Flop

/*********************************************************************/
/*                                                                   */
/*     Model transformation modifications:                           */
/*                                                                   */
/*          1) CDL decoder structure nonexistent in ISP' and un-     */
/*     necessary for model. Eliminated.                              */
/*          2) Multi-phase clock structure nonexistent in ISP'.      */
/*     Operations on registers will provide its equivalent.          */
/*          3) Switch structure nonexistent in ISP'. Operation on a  */
/*     register will provide its equivalent.                         */
/*          4) The declared bus structures are modeled with registers */
/*     without loss of model accurracy. This done to maintain model  */
/*     equivalency and simplicity.                                   */
/*          5) The memory word length was reduced from 16 to 8 bit   */
/*     words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*     their PC incrementation, and to enable the use of existing    */
/*     MC68000 assembler and linker/loader models. The memory was    */
/*     also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                   */
/*********************************************************************/

IABUS<31:0>,               ! Internal Address Bus
IDBUS<31:0>,               ! Internal Data Bus
twait<7:0>,                ! Wait Cycle Counter
SWITCH,                 ! Power Switch
PHI1,                   ! Phase 1 Of Two-Phase Clock
PHI2;                   ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                   */
/*               External Address and Data Bus                       */
/*                                                                   */
/*********************************************************************/

DBUS<15:0>,                ! External Data Bus
```

```
ABUS<23:1>;                    ! External Address Bus(changed)

format

/**********************************************************************/
/*                                                                    */
/*                       Register Subfields                           */
/*                                                                    */
/**********************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/**********************************************************************/
/*                                                                    */
/*                  16K 16-Bit Word Internal Memory                   */
/*                                                                    */
/**********************************************************************/

M[0:32767]<7:0>;

macro

/**********************************************************************/
/*                                                                    */
/*                     Logic Level Macros                             */
/*                                                                         */
/***************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/***************************************************************************/
/*                                                                         */
/*  Power On and Initialization. This process was not modeled but is      */
/*  added to initialize signals and registers.                             */
/*                                                                         */
/***************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                        ! Turn Power On
        next;                               ! Execute Assignment
        READY = lo;                         ! System Not Ready
        RESET = lo;                         ! Assert Reset For
        delay(100);                         ! 100 Miliseconds(Active Low)
        RESET = hi;                         ! Deactivate Reset
        next;                               ! Execute Pending Assignments
        ASN = hi;                           ! Initialize Address Strobe
        LDSN = hi;                          ! Initialize Lower Data Strobe
        UDSN = hi;                          ! Initialize Upper Data Strobe
        DTACKN = hi;                        ! Initialize Data Transfer Acknowledge
        RW = hi;                            ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                     ! Place Data Bus In High Impedance State
        M[0x100e] = 0xff;                   ! Place Memory Locations Following The
        M[0x100f] = 0xff;                     ! JMP Instruction In A High State
        HALT = hi;                          ! Initialize Halt Flip-Flop(Active
                                            ! Low)
        T = 0;                              ! Initialize Clock Cycle Counter
        READY = hi;                         ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*     Routine Initialization Per Hamby and Guillory             */
        /*                                                               */
        /*****************************************************************/
        A[0] = 0x1004;                      ! Place Hex 1004 Into A[0]
        PC = 0x1000;                        ! Place Hex 1000 Into Program Counter
        next                                ! Execute Assignments
        )

/***************************************************************************/
/*                                                                         */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary      */
/*  to retrieve modeled instructions for simulation and analysis. It      */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory    */
/*  on page VI-15 of their thesis.                                         */
/*                                                                  */
/********************************************************************/

fetch_initial_instruction :=
      (

      /**************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /**************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /**************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
      while DTACKN eql hi                     ! Wait For Memory To Place
            (                                 ! Data On The Bus
            next;                             ! Execute Impending Assignments
```

```
        PHI1 = lo;                          ! Phase 2
        PHI2 = hi;                          ! Of Clock Cycle 2
        next;                               ! Execute Assignments

        /*******************************************************/
        T = 3;                              ! Clock Cycle 3
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1
        PHI2 = lo;                          ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
        DTACKN = lo;                        ! Asserts DTACKN(Added)
        next;                               ! Execute Pending Assignments

        /*******************************************************/
        T = 2                               ! Return To  Phase 2
                                            ! Of Clock Cycle 2
        );
        next;                               ! Execute Impending Assignments

    /************************************************************/
    T = 3;                                  ! Clock Cycle 3
    next;                                   ! Execute Assignment

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 3
    EXDBUF = DBUS;                          ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
    next;                                   ! Execute Pending Assignments

    /************************************************************/
    T = 4;                                  ! Clock Cycle 4
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 4
    PFR = EXDBUF;                           ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
    next;                                   ! Execute Pending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 4
    ASN = hi;                               ! Deactivate Address Strobe
    LDSN = hi;                              ! Deactivate Lower Data Strobe
    UDSN = hi;                              ! Deactivate Upper Data Strobe
    IR = PFR;                               ! Contents Of Prefetch Register
                                            ! Are Placed Into Instruction
                                            ! Register
    DTACKN = hi;                            ! Deactivate Data Transfer(Added)
```

```
                                          ! Acknowledge
     PC = PC + 4;                         ! Increment Program Counter
     next;                                ! Execute Pending Assignments
     T = 0                                ! Reset Clock Cycle Counter
     )

andi :=                                   ! AND.W #$DFFF,SR
    (
    SRMODE = lo;                          ! Effect Of Instruction
    IR<15:8> = M[PC];                     ! Prefetch Next Instruction
    IR<7:0> = M[PC + 1];
    next;                                 ! Is To Set Status Register
       PC = PC + 2;                         ! Increment Program Counter
    T = 5;                                ! Supervisor Bit To User
    next;                                 ! Mode
    T = 0                                 ! Requires 6 Clock Cycles
    )

move :=                                   ! MOVE.W #$5555,D1
     (

     /*****************************************************************/

     PHI1 = hi;                           ! Phase 1 Of
     PHI2 = lo;                           ! Clock Cycle 0
     DBUS = 0xffff;                       ! Place Data Bus In High Impedance
     RW = hi;                             ! Memory Read
     ADENABLE = lo;                       ! Disable Address Bus Buffer
     ABUS = 0xffffff;                     ! Address Bus High Impedanced
     DBENABLE = lo;                       ! Disable Data Bus Buffer
     IABUS<31:1> = PC<31:1>;              ! Place PC On Internal Address
                                          ! Bus
     next;                                ! Execute Pending Assignments

     PHI1 = lo;                           ! Phase 2 Of
     PHI2 = hi;                           ! Clock Cycle 0
     ADENABLE = hi;                       ! Enable Address Bus Buffer
     EXABUF = IABUS<23:1>;                ! Gate Internal Address Bus
                                          ! Into External Address Buffer
     FCMODE = SRMODE;                     ! User Mode
     FCSPACE = 2;                         ! Accessing Program
     SRCARRY = lo;                        ! Clear Status Register Carry Bit
     SROVER = lo;                         ! Clear Status Register Overflow Bit
     SRZERO = lo;                         ! Clear Status Register Zero Bit
     SRNEG  = lo;                         ! Clear Status Register Negative Bit
     ABUS = IABUS<23:1>;                  ! Place PC On Address Bus (Added)
     next;                                ! Execute Impending Assignments

     /*****************************************************************/
     T = 1;                               ! Clock Cycle 1
     next;                                ! Execute Assignment

     PHI1 = hi;                           ! Phase 1 Of
```

```
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*******************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*******************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments
```

```
/****************************************************************/
T = 4;                                    ! Clock Cycle 4
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 4
IDBUS = EXDBUF;
if EXDBUF eql 0                           ! Set Status Register
   SRZERO = hi;                           ! Bits As Appropriate
if EXDBUF<15> eql 1
   SRNEG = hi;
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 4
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
UDSN = hi;                                ! Deactivate Upper Data Strobe
PC = PC + 2;                              ! Increment PC
DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
D[1] = IDBUS;                             ! Place Contents Of Internal
                                          ! Data Bus Into D[2]
next;                                     ! Execute Impending Assignments

/****************************************************************/

T = 5;                                    ! Clock Cycle 5
next;

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 5
DBUS = 0xffff;                            ! Place Data Bus In High Impedance
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
ABUS = 0xffffff;                          ! Address Bus High Impedanced
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;                   ! Place PC On Internal Address
                                          ! Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 5
ADENABLE = hi;                            ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                     ! Gate Internal Address Bus
                                          ! Into External Address Buffer
FCMODE = SRMODE;                          ! User Mode
FCSPACE = 2;                              ! Accessing Program
ABUS = IABUS<23:1>;                       ! Place PC On Address Bus
next;                                     ! Execute Impending Assignments

/****************************************************************/
T = 6;                                    ! Clock Cycle 6
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 7
    next;                               ! Execute Assignments

    /*********************************************************/
    T = 8;                              ! Clock Cycle 8
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 8
    DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
    DTACKN = lo;                        ! Asserts DTACKN(Added)
    next;                               ! Execute Pending Assignments

    /*********************************************************/
    T = 7                               ! Return To  Phase 2
                                        ! Of Clock Cycle 7
    );
    next;                               ! Execute Impending Assignments

/*************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
```

```
                                                    ! Bus Buffer
        next;                                       ! Execute Pending Assignments

        /**************************************************************/
        T = 9;                                      ! Clock Cycle 9
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1
        PHI2 = lo;                                  ! Of Clock Cycle 9
        PFR = EXDBUF;                               ! The Contents Of The External
                                                    ! Data Bus Buffer Are Placed
                                                    ! In Prefetch Register
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 9
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        PC = PC + 2;                                ! Increment Program Counter
                                                    ! Are Placed Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                                    ! Acknowledge
        IR = PFR;                                   ! Load Instruction Register With
                                                    ! Prefetch Register
        next;

        T = 0
        )

    jmp :=                                          ! JMP (A0)
        (

        /**************************************************************/

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 0
        DBUS = 0xffff;                              ! Place Data Bus In A High Impedance
        RW = hi;                                    ! Memory Read
        ADENABLE = lo;                              ! Disable Address Bus Buffer
        DBENABLE = lo;                              ! Disable Data Bus Buffer
        IABUS = PC;                                 ! Place PC On Internal Address
                                                    ! Bus
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2 Of
        PHI2 = hi;                                  ! Clock Cycle 0
        ADENABLE = hi;                              ! Enable Address Bus Buffer
        EXABUF = IABUS;                             ! Gate Internal Address Bus
                                                    ! Into External Address Buffer
        FCMODE = SRMODE;                            ! User Mode
        FCSPACE = 2;                                ! Accessing Program
```

```
next;                                   ! Execute Pending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***********************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                                ! Execute Impending Assignments

   /******************************************************************/
   T = 3;                                  ! Clock Cycle 3
   next;                                   ! Execute Assignment

   PHI1 = lo;                              ! Phase 2
   PHI2 = hi;                              ! Of Clock Cycle 3
   EXDBUF = DBUS;                          ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
   next;                                   ! Execute Pending Assignments

   /******************************************************************/
   T = 4;                                  ! Clock Cycle 4
   next;                                   ! Execute Assignment

   PHI1 = hi;                              ! Phase 1
   PHI2 = lo;                              ! Of Clock Cycle 4
   next;
   PFR = EXDBUF;                           ! The Contents Of The External
                                           ! Data Bus Buffer Are Placed
                                           ! In Prefetch Register
   next;                                   ! Execute Pending Assignments

   PHI1 = lo;                              ! Phase 2
   PHI2 = hi;                              ! Of Clock Cycle 4
   ASN = hi;                               ! Deactivate Address Strobe
   LDSN = hi;                              ! Deactivate Lower Data Strobe
   UDSN = hi;                              ! Deactivate Upper Data Strobe
   DTACKN = hi;                            ! Deactivate Data Transfer
                                           ! Acknowledge(Added)
   next;
   /******************************************************************/
   T = 5;                                  ! Clock Cycle 5
   next;                                   ! Execute Previous Assignment

   PHI1 = hi;                              ! Phase 1 Of
   PHI2 = lo;                              ! Clock Cycle 5
   RW = hi;                                ! Memory Read
   ADENABLE = lo;                          ! Disable Address Bus Buffer
   DBENABLE = lo;                          ! Disable Data Bus Buffer
   IABUS = PC;                             ! Place PC On Internal Address
                                           ! Bus
   next;                                   ! Execute Pending Assignments

   PHI1 = lo;                              ! Phase 2 Of
   PHI2 = hi;                              ! Clock Cycle 5
   ADENABLE = hi;                          ! Enable Address Bus Buffer
   FCMODE = SRMODE;                        ! User Mode
   FCSPACE = 2;                            ! Accessing Program
   EXABUF = IABUS;                         ! Gate Internal Address Bus
   next;                                   ! Into External Address Buffer
```

```
ABUS = EXABUF;                         ! Address Placed On Bus(Added)
next;                                  ! Execute Pending Assignments

/*******************************************************************/
T = 6;                                 ! Clock Cycle 6
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 6
ASN = lo;                              ! Assert Address Strobe
LDSN = lo;                             ! Assert Lower Data Strobe
UDSN = lo;                             ! Assert Upper Data Strobe
DBENABLE = hi;                         ! Enable Data Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 6
next;                                  ! Execute Pending Assignments

/*******************************************************************/
T = 7;                                 ! Clock Cycle 7
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 7
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 7
      next;                            ! Execute Assignments

      /*************************************************************/
      T = 8;                           ! Clock Cycle 8
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /*************************************************************/
      T = 7                           ! Return To Phase 2
                                      ! Of Clock Cycle 7
      );
      next;                           ! Execute Impending Assignments

/*******************************************************************/
T = 8;                                 ! Clock Cycle 8
next;                                  ! Execute Assignment
```

```
        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 8
        EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                    ! Is Placed In External Data
                                                    ! Bus Buffer
        next;                                       ! Execute Pending Assignments

        /**************************************************************/
        T = 9;                                      ! Clock Cycle 9
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1
        PHI2 = lo;                                  ! Of Clock Cycle 9
        PFR = EXDBUF;                               ! The Contents Of The External
                                                    ! Data Bus Buffer Are Placed
                                                    ! In Prefetch Register
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2
        PHI2 = hi;                                  ! Of Clock Cycle 9
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        PC = PC + 2;                                ! Increment Program Counter
        IR = PFR;                                   ! Place Contents Of Prefetch
                                                    ! Register Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer
                                                    ! Acknowledge(Added)
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                          (
                          case IR
                               0x323c: move     ! MOVE.W #$5555,D1
                               0x027c: andi     ! AND.W #$DFFF,SR
                               047320: jmp      ! JMP (A0) If IR = Octal Value
                          esac
                          )

main :=
      (
      power_on_initialize;
      fetch_initial_instruction;
      while READY eql hi
           (
           decode_execute_prefetch
           )
      )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE ADD.W D3,D5 INSTRUCTION             */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                      Structure Declarations                           */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                 M68000 Programming Registers                          */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,                ! 8 Data Registers
A[0:6]<31:0>,                ! 7 Address Registers
UA7<31:0>,                   ! User Stack Pointer
SA7<31:0>,                   ! System Stack Pointer
PC<31:0>,                    ! Program Counter
SR<15:0>,                    ! Status Register

/*************************************************************************/
/*                                                                       */
/*                 Temporary Internal Registers                          */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                   ! Prefetch Register
IR<15:0>,                    ! Instruction Register
FC<2:0>,                     ! Function Code Register
EXDBUF<15:0>,                ! External Data Bus Buffer Register
EXABUF<23:1>,                ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,               ! ALU Buffer 1
ALUBUF2<31:0>,               ! ALU Buffer 2
DTEMP<15:0>,                 ! Temporary Data Storage
DISREG<31:0>,                ! Temporary Displacement Storage
SRTEMP<15:0>,                ! Temporary Status Register Storage
                             ! (Exception Processing)
IRTEMP<15:0>,                ! Temporary Instruction Register Storage
                             ! (Exception Processing)
TEMPADR<31:0>,               ! Temporary Cycle Address Storage
                             ! (Exception Processing)
ACTYPE<15:0>,                ! Temporary Access Type Storage
                             ! (Exception Processing)
VECADR<23:0>,                ! Temporary Vector Address Storage
                             ! (Exception Processing)
```

```
HANADR<31:0>,                ! Temporary Address Storage For
                             ! Exception Handler Routine
T<7:0>,                      ! Clock Cycle Counter
RESET,                    ! Reset Flip-Flop
HALT,                     ! Halt Flip-Flop
RW,                       ! Read/Write Flip-Flop
ADENABLE,                 ! Address Bus Buffer Enable
DBENABLE,                 ! Data Bus Buffer Enable
ASN,                      ! Address Strobe Flip-Flop
LDSN,                     ! Lower Data Strobe Flip-Flop
UDSN,                     ! Upper Data Strobe Flip-Flop
DTACKN,                   ! Data Transfer Acknowledge Flip-Flop
COUT,                     ! Carry Flip-Flop
EXCEPT,                   ! Exception Processing Flip-Flop
READY,                    ! Ready Flip-Flop

/***********************************************************************/
/*                                                                     */
/*      Model transformation modifications:                            */
/*                                                                     */
/*           1) CDL decoder structure nonexistent in ISP' and un-      */
/*      necessary for model. Eliminated.                               */
/*           2) Multi-phase clock structure nonexistent in ISP'.       */
/*      Operations on registers will provide its equivalent.           */
/*           3) Switch structure nonexistent in ISP'. Operation on a   */
/*      register will provide its equivalent.                          */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model   */
/*      equivalency and simplicity.                                    */
/*           5) The memory word length was reduced from 16 to 8 bit    */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing     */
/*      MC68000 assembler and linker/loader models. The memory was     */
/*      also reduced from 8 Mwords to 32 Kbytes.                       */
/*                                                                     */
/***********************************************************************/

IABUS<31:0>,                 ! Internal Address Bus
IDBUS<31:0>,                 ! Internal Data Bus
SWITCH,                   ! Power Switch
PHI1,                     ! Phase 1 Of Two-Phase Clock
PHI2;                     ! Phase 2 Of Two-Phase Clock

port

/***********************************************************************/
/*                                                                     */
/*              External Address and Data Bus                          */
/*                                                                     */
/***********************************************************************/

DBUS<15:0>,                  ! External Data Bus
ABUS<23:1>;                  ! External Address Bus(changed)
```

```
format

/***************************************************************************/
/*                                                                         */
/*                       Register Subfields                                */
/*                                                                         */
/***************************************************************************/

PCADDR       = PC<23:0>,      ! Program Counter Address Field
SRTRACE      = SR<15>,        ! Trace Bit
SRMODE       = SR<13>,        ! Mode Selection Bit
SRCARRY      = SR<0>,         ! Carry Bit
SROVER       = SR<1>,         ! Overflow Bit
SRZERO       = SR<2>,         ! Zero Bit
SRNEG        = SR<3>,         ! Negative Bit
SREX         = SR<4>,         ! Extend Bit
SRMASK       = SR<10:8>,      ! Interrupt Mask
FCSPACE      = FC<1:0>,       ! Memory Access Address Space
FCMODE       = FC<2>,         ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,      ! PC Low Word
PCHI         = PC<31:16>,     ! PC High Word
D0LWORD      = D[0]<15:0>,    ! D[0] Low Word
D1LWORD      = D[1]<15:0>,    ! D[1] Low Word
D2LWORD      = D[2]<15:0>,    ! D[2] Low Word
D3LWORD      = D[3]<15:0>,    ! D[3] Low Word
D4LWORD      = D[4]<15:0>,    ! D[4] Low Word
D5LWORD      = D[5]<15:0>,    ! D[5] Low Word
D6LWORD      = D[6]<15:0>,    ! D[6] Low Word
D7LWORD      = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***************************************************************************/
/*                                                                         */
/*                   16K 16-Bit Word Internal Memory                       */
/*                                                                         */
/***************************************************************************/

M[0:32767]<7:0>;

macro

/***************************************************************************/
/*                                                                         */
/*                       Logic Level Macros                                */
/*                                                                         */
```

```
/**********************************************************************/

lo    = 0 %,
hi    = 1 %,
off   = 0 %,
on    = 1 %,
clear = 0 %;

/**********************************************************************/
/*                                                                    */
/*  Power On and Initialization. This process was not modeled but is */
/*  added to initialize signals and registers.                        */
/*                                                                    */
/**********************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                  ! Place Data Bus In High Impedance State
        M[0x1008] = 0xff;               ! Place Memory Locations Following The
        M[0x1009] = 0xff;               ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /**************************************************************/
        /*                                                            */
        /*     Routine Initialization Per Hamby and Guillory          */
        /*                                                            */
        /**************************************************************/
        SRMODE = lo;                    ! Set Status Register To User Mode
        D[1] = 0x00000003;              ! Place Hex 00000003 Into D[1]
        D[3] = 0x00000002;              ! Place Hex 00000002 Into D[2]
        D[5] = 0x00000000;              ! Initialize D[5] To Zero
        A[0] = 0x1000;                  ! Place Hex 1000 Into A[0]
        A[2] = 0x2000;                  ! Store Data At Hex 2000
        PC = 0x1000;                    ! Place Hex 1000 Into Program Counter
        next                            ! Execute Assignments
        )

/**********************************************************************/
/*                                                                    */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                     */
/*                                                                     */
/***********************************************************************/

fetch_initial_instruction :=
      (

      /*************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /*************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /*************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
```

```
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /*******************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /*******************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
```

```
        IR = PFR;                                   ! Contents Of Prefetch Register
                                                    ! Are PlaceJ Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                                    ! Acknowledge
        PC = PC + 2;                                ! Increment Program Counter
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

  add :=                                            ! ADD.W D3,D5
        (

        /*****************************************************************/

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 0
        DBUS = 0xffff;                              ! Place Data Bus In High Impedance
        RW = hi;                                    ! Memory Read
        IDBUS = D3LWORD;                            ! Place Low Word Of D[3]
                                                    ! Onto Internal Data Bus
        ADENABLE = lo;                              ! Disable Address Bus Buffer
        DBENABLE = lo;                              ! Disable Data Bus Buffer
        IABUS = PC;                                 ! Place PC On Internal Address
                                                    ! Bus
        next;                                       ! Execute Pending Assignments

        PHI1 = lo;                                  ! Phase 2 Of
        PHI2 = hi;                                  ! Clock Cycle 0
        ADENABLE = hi;                              ! Enable Address Bus Buffer
        EXABUF = IABUS;                             ! Gate Internal Address Bus
                                                    ! Into External Address Buffer
        FCMODE = SRMODE;                            ! User Mode
        FCSPACE = 2;                                ! Accessing Program
        ALUBUF1 = IDBUS;                            ! Place Data From Internal Data Bus
                                                    ! Into ALU Buffer 1
        next;                                       ! Execute Impending Assignments
        ABUS = EXABUF;                              ! Address Placed On Bus(Added)
        next;                                       ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                      ! Clock Cycle 1
        next;                                       ! Execute Assignment

        PHI1 = hi;                                  ! Phase 1 Of
        PHI2 = lo;                                  ! Clock Cycle 1
         3N = lo;                                   ! Assert Address Strobe
        LDSN = lo;                                  ! Assert Lower Data Strobe
        UDSN = lo;                                  ! Assert Upper Data Strobe
        DBENABLE = hi;                              ! Enable Data Bus
        IDBUS = D5LWORD;                            ! Place Low Word From D[5]
                                                    ! Onto Internal Data Bus
        next;                                       ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
ALUBUF2 = IDBUS;                        ! Place Data From Internal Data
                                        ! Bus Into ALU Buffer 2
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
IDBUS = ALUBUF1 + ALUBUF2;              ! Place Sum From ALU Onto
                                        ! Internal Data Bus
SRCARRY = lo;                           ! Reset Condition Codes
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
SREX = lo;
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
```

```
                                          ! Bus Buffer
    D5LWORD = IDBUS;                      ! Sum On Internal Data Bus
                                          ! Is Place Into Low Word Of D[5]
    next;                                 ! Execute Pending Assignments

    /****************************************************************/
    T = 4;                                ! Clock Cycle 4
    next;                                 ! Execute Assignment

    PHI1 = hi;                            ! Phase 1
    PHI2 = lo;                            ! Of Clock Cycle 4
    if D5LWORD eql 0                      ! Set Condition Code
       SRZERO = hi;
    if COUT eql 1                         ! Bits As Appropriate
       (
       SRCARRY = hi;
       SREX = hi
       );
    if D[5]<15>
       SRNEG = hi;
    SROVER = ((not D[5]<15>) and ALUBUF1<15> and ALUBUF2<15>)
             or (D[5]<15> and (not ALUBUF1<15>) and (not ALUBUF2<15>));
    PFR = EXDBUF;                         ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
    next;                                 ! Execute Pending Assignments

    PHI1 = lo;                            ! Phase 2
    PHI2 = hi;                            ! Of Clock Cycle 4
    ASN = hi;                             ! Deactivate Address Strobe
    LDSN = hi;                            ! Deactivate Lower Data Strobe
    UDSN = hi;                            ! Deactivate Upper Data Strobe
    IR = PFR;                             ! Contents Of Prefetch Register
                                          ! Are Placed Into Instruction
                                          ! Register
    DTACKN = hi;                          ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
    PC = PC + 2;                          ! Increment Program Counter
    next;                                 ! Execute Impending Assignments
    T = 0                                 ! Reset Clock Cycle Counter
    )

moved :=                                   ! MOVE.W D1,D5
    (

    /****************************************************************/

    PHI1 = hi;                            ! Phase 1 Of
    PHI2 = lo;                            ! Clock Cycle 0
    DBUS = 0xffff;                        ! Place Data Bus In High Impedance
    RW = hi;                              ! Memory Read
    ADENABLE = lo;                        ! Disable Address Bus Buffer
    DBENABLE = lo;                        ! Disable Data Bus Buffer
```

```
IABUS = PC;                     ! Place PC On Internal Address
                                ! Bus
IDBUS = D1LWORD;                ! Place Low Word From D[1] Onto
                                ! Internal Data Bus
next;                           ! Execute Pending Assignments

PHI1 = lo;                      ! Phase 2 Of
PHI2 = hi;                      ! Clock Cycle 0
ADENABLE = hi;                  ! Enable Address Bus Buffer
EXABUF = IABUS;                 ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRMODE;                ! User Mode
FCSPACE = 2;                    ! Accessing Program
SRCARRY = lo;                   ! Clear Status Register Carry Bit
SROVER = lo;                    ! Clear Status Register Overflow Bit
SRZERO = lo;                    ! Clear Status Register Zero Bit
SRNEG  = lo;                    ! Clear Status Register Negative Bit
D5LWORD = IDBUS;                ! Place Data From Internal Data Bus
                                ! Into Low Word Of D[2]
next;                           ! Execute Impending Assignments
ABUS = EXABUF;                  ! Address Placed On Bus(Added)
next;                           ! Execute Pending Assignments

/*****************************************************************/
T = 1;                          ! Clock Cycle 1
next;                           ! Execute Assignment

PHI1 = hi;                      ! Phase 1 Of
PHI2 = lo;                      ! Clock Cycle 1
ASN = lo;                       ! Assert Address Strobe
LDSN = lo;                      ! Assert Lower Data Strobe
UDSN = lo;                      ! Assert Upper Data Strobe
DBENABLE = hi;                  ! Enable Data Bus
if D5LWORD eql 0                ! Set Status Register Zero Bit
   SRZERO = hi;                 ! If Moved Data Is Zero
next;                           ! Execute Pending Assignments

PHI1 = lo;                      ! Phase 2
PHI2 = hi;                      ! Of Clock Cycle 1
if D[5]<15>                     ! Set Status Register Negative
   SRNEG = hi;                  ! Bit If Moved Data Is Negative
next;                           ! Execute Pending Assignments

/*****************************************************************/
T = 2;                          ! Clock Cycle 2
next;                           ! Execute Assignment

PHI1 = hi;                      ! Phase 1
PHI2 = lo;                      ! Of Clock Cycle 2
while DTACKN eql hi             ! Wait For Memory To Place
      (                         ! Data On The Bus
      next;                     ! Execute Impending Assignments
```

```
          PHI1 = lo;                     ! Phase 2
          PHI2 = hi;                     ! Of Clock Cycle 2
          next;                          ! Execute Assignments

          /*************************************************************/
          T = 3;                         ! Clock Cycle 3
          next;                          ! Execute Assignment

          PHI1 = hi;                     ! Phase 1
          PHI2 = lo;                     ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];          ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];       ! On Data Bus And
          DTACKN = lo;                   ! Asserts DTACKN(Added)
          next;                          ! Execute Pending Assignments

          /*************************************************************/
          T = 2                          ! Return To  Phase 2
                                         ! Of Clock Cycle 2
          );
          next;                          ! Execute Impending Assignments

      /*******************************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 3
      EXDBUF = DBUS;                     ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
      next;                              ! Execute Pending Assignments

      /*******************************************************************/
      T = 4;                             ! Clock Cycle 4
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 4
      PFR = EXDBUF;                      ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
      next;                              ! Execute Pending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 4
      ASN = hi;                          ! Deactivate Address Strobe
      LDSN = hi;                         ! Deactivate Lower Data Strobe
      UDSN = hi;                         ! Deactivate Upper Data Strobe
      IR = PFR;                          ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
      DTACKN = hi;                       ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
```

```
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

movei :=                                         ! MOVE.W D5,(A2)
        (

        /******************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Impending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /******************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        DBENABLE = hi;                          ! Enable Data Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 1
        next;                                   ! Execute Pending Assignments

        /******************************************************************/
        T = 2;                                  ! Clock Cycle 2
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 2
```

```
while DTACKN eql hi                    ! Wait For Memory To Place
      (                                ! Data On The Bus
      next;                            ! Execute Impending Assignments

      PHI1 = lo;                       ! Phase 2
      PHI2 = hi;                       ! Of Clock Cycle 2
      next;                            ! Execute Assignments

      /**********************************************************/
      T = 3;                           ! Clock Cycle 3
      next;                            ! Execute Assignment

      PHI1 = hi;                       ! Phase 1
      PHI2 = lo;                       ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];            ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
      DTACKN = lo;                     ! Asserts DTACKN(Added)
      next;                            ! Execute Pending Assignments

      /**********************************************************/
      T = 2                            ! Return To  Phase 2
                                       ! Of Clock Cycle 2
      );
      next;                            ! Execute Impending Assignments

/****************************************************************/
T = 3;                                 ! Clock Cycle 3
next;                                  ! Execute Assignment

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 3
EXDBUF = DBUS;                         ! Instruction On Data Bus
                                       ! Is Placed In External Data
                                       ! Bus Buffer
next;                                  ! Execute Pending Assignments

/****************************************************************/
T = 4;                                 ! Clock Cycle 4
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 4
PFR = EXDBUF;                          ! The Contents Of The External
                                       ! Data Bus Buffer Are Placed
                                       ! In Prefetch Register
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 4
ASN = hi;                              ! Deactivate Address Strobe
LDSN = hi;                             ! Deactivate Lower Data Strobe
UDSN = hi;                             ! Deactivate Upper Data Strobe
                                       ! Are Placed Into Instruction
```

```
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;

/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBUS = 0xffff;                          ! Return Data Bus To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = A[2];                           ! Place A[2] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
IDBUS = D5LWORD;                        ! Place Low Word from D[5] On
                                        ! Internal Data Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                         ! Place Contents Of Internal
                                        ! Data Bus Into External Data Buffer
SRCARRY = lo;                           ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
if EXDBUF eql 0                         ! Set Zero Condition Bit If Needed
   SRZERO = hi;
DBUS = EXDBUF;                          ! Place Data On External Data Bus
```

```
DBENABLE = hi;                        ! Enable Data Bus
next;                                 ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                ! Clock Cycle 7
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 7
if EXDBUF<15>                         ! Set Negative Condition Bit
   SRNEG = hi;                        ! If Needed
UDSN = lo;                            ! Activate Upper And
LDSN = lo;                            ! Lower Data Strobes
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 7
      next;                           ! Execute Assignments

      /**********************************************************/
      T = 8;                          ! Clock Cycle 8
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 8
      M[ABUS] = DBUS<15:8>;           ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;        ! In Memory
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /**********************************************************/
      T = 7                          ! Return To Phase 2
                                     ! Of Clock Cycle 7
      );
      next;                          ! Execute Impending Assignments

/*****************************************************************/
T = 8;                                ! Clock Cycle 8
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 8
next;                                 ! Execute Pending Assignments

/*****************************************************************/
T = 9;                                ! Clock Cycle 9
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 9
next;                                 ! Execute Pending Assignments
```

```
        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0
        )


Jmp :=                                          ! JMP (A0)
        (

        /*********************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Data Bus High Impedanced
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /*********************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
```

```
IABUS = A[0];                                ! Move Jump Address From A[0]
                                             ! To Internal Address Buffer
DBENABLE = hi;                               ! Enable Data Bus
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 1
PC = IABUS;                                  ! Place Jump Address Into Program
                                             ! Counter
next;

/*************************************************************/
T = 2;                                       ! Clock Cycle 2
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1
PHI2 = lo;                                   ! Of Clock Cycle 2
while DTACKN eql hi                          ! Wait For Memory To Place
      (                                      ! Data On The Bus
      next;                                  ! Execute Impending Assignments

      PHI1 = lo;                             ! Phase 2
      PHI2 = hi;                             ! Of Clock Cycle 2
      next;                                  ! Execute Assignments

      /*******************************************************/
      T = 3;                                 ! Clock Cycle 3
      next;                                  ! Execute Assignment

      PHI1 = hi;                             ! Phase 1
      PHI2 = lo;                             ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                  ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];               ! On Data Bus And
      DTACKN = lo;                           ! Asserts DTACKN(Added)
      next;                                  ! Execute Pending Assignments

      /*******************************************************/
      T = 2                                  ! Return To Phase 2
                                             ! Of Clock Cycle 2
      );
      next;                                  ! Execute Impending Assignments

/*************************************************************/
T = 3;                                       ! Clock Cycle 3
next;                                        ! Execute Assignment

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 3
EXDBUF = DBUS;                               ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
next;                                        ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
```

```
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/**********************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 7
    next;                               ! Execute Assignments

    /*****************************************************************/
    T = 8;                              ! Clock Cycle 8
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 8
    DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
    DTACKN = lo;                        ! Asserts DTACKN(Added)
    next;                               ! Execute Pending Assignments

    /*****************************************************************/
    T = 7                              ! Return To Phase 2
                                       ! Of Clock Cycle 7
    );
    next;                              ! Execute Impending Assignments

/**********************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/**********************************************************************/
T = 9;                                  ! Clock Cycle 9
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
```

```
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register

        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                        (
                        case IR
                             0155103: add    ! ADD.W D3,D5
                             035001 : moved  ! MOVE.W D1,D5
                             032205 : movei  ! MOVE.W D5,(A2)
                             047320 : jmp    ! JMP (A0) If IR = Octal Value
                        esac
                        )

main :=
      (
      power_on_initialize;
      fetch_initial_instruction;
      while READY eql hi
            (
            decode_execute_prefetch
            )
      )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE BEQ $1000    INSTRUCTION            */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                    Structure Declarations                             */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                 M68000 Programming Registers                          */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/*************************************************************************/
/*                                                                       */
/*                 Temporary Internal Registers                          */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,                 ! Temporary Address Storage For
                              ! Exception Handler Routine
T<7:0>,                       ! Clock Cycle Counter
RESET,                     ! Reset Flip-Flop
HALT,                      ! Halt Flip-Flop
RW,                        ! Read/Write Flip-Flop
ADENABLE,                  ! Address Bus Buffer Enable
DBENABLE,                  ! Data Bus Buffer Enable
ASN,                       ! Address Strobe Flip-Flop
LDSN,                      ! Lower Data Strobe Flip-Flop
UDSN,                      ! Upper Data Strobe Flip-Flop
DTACKN,                    ! Data Transfer Acknowledge Flip-Flop
COUT,                      ! Carry Flip-Flop
EXCEPT,                    ! Exception Processing Flip-Flop
READY,                     ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                           */
/*                                                                    */
/*           1) CDL decoder structure nonexistent in ISP' and un-     */
/*      necessary for model. Eliminated.                              */
/*           2) Multi-phase clock structure nonexistent in ISP'.      */
/*      Operations on registers will provide its equivalent.          */
/*           3) Switch structure nonexistent in ISP'. Operation on a  */
/*      register will provide its equivalent.                         */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model  */
/*      equivalency and simplicity.                                   */
/*           5) The memory word length was reduced from 16 to 8 bit   */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing    */
/*      MC68000 assembler and linker/loader models. The memory was    */
/*      also reduced from 8 Mwords to 32 Kbytes.                      */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,                  ! Internal Address Bus
IDBUS<31:0>,                  ! Internal Data Bus
SWITCH,                    ! Power Switch
PHI1,                      ! Phase 1 Of Two-Phase Clock
PHI2;                      ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*              External Address and Data Bus                         */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                   ! External Data Bus
ABUS<23:1>;                   ! External Address Bus(changed)
```

```
format

/***********************************************************************/
/*                                                                     */
/*                        Register Subfields                           */
/*                                                                     */
/***********************************************************************/

PCADDR       = PC<23:0>,        ! Program Counter Address Field
SRTRACE      = SR<15>,          ! Trace Bit
SRMODE       = SR<13>,          ! Mode Selection Bit
SRCARRY      = SR<0>,           ! Carry Bit
SROVER       = SR<1>,           ! Overflow Bit
SRZERO       = SR<2>,           ! Zero Bit
SRNEG        = SR<3>,           ! Negative Bit
SREX         = SR<4>,           ! Extend Bit
SRMASK       = SR<10:8>,        ! Interrupt Mask
FCSPACE      = FC<1:0>,         ! Memory Access Address Space
FCMODE       = FC<2>,           ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,        ! PC Low Word
PCHI         = PC<31:16>,       ! PC High Word
D0LWORD      = D[0]<15:0>,      ! D[0] Low Word
D1LWORD      = D[1]<15:0>,      ! D[1] Low Word
D2LWORD      = D[2]<15:0>,      ! D[2] Low Word
D3LWORD      = D[3]<15:0>,      ! D[3] Low Word
D4LWORD      = D[4]<15:0>,      ! D[4] Low Word
D5LWORD      = D[5]<15:0>,      ! D[5] Low Word
D6LWORD      = D[6]<15:0>,      ! D[6] Low Word
D7LWORD      = D[7]<15:0>,      ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/***********************************************************************/
/*                                                                     */
/*                  16K 16-Bit Word Internal Memory                    */
/*                                                                     */
/***********************************************************************/

M[0:32767]<7:0>;

macro

/***********************************************************************/
/*                                                                     */
/*                        Logic Level Macros                           */
/*                                                                     */
```

```
/*************************************************************************/

lo    = 0 %,
hi    = 1 %,
off   = 0 %,
on    = 1 %,
clear = 0 %;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                      ! Turn Power On
        next;                             ! Execute Assignment
        READY = lo;                       ! System Not Ready
        RESET = lo;                       ! Assert Reset For
        delay(100);                       ! 100 Miliseconds(Active Low)
        RESET = hi;                       ! Deactivate Reset
        next;                             ! Execute Pending Assignments
        ASN = hi;                         ! Initialize Address Strobe
        LDSN = hi;                        ! Initialize Lower Data Strobe
        UDSN = hi;                        ! Initialize Upper Data Strobe
        DTACKN = hi;                      ! Initialize Data Transfer Acknowledge
        RW = hi;                          ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                   ! Place Data Bus In High Impedance State
        M[0x100a] = 0xff;                 ! Place Memory Locations Following The
        M[0x100b] = 0xff;                   ! JMP Instruction In A High State
        HALT = hi;                        ! Initialize Halt Flip-Flop(Active
                                          ! Low)
        T = 0;                            ! Initialize Clock Cycle Counter
        READY = hi;                       ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*     Routine Initialization Per Hamby and Guillory             */
        /*                                                               */
        /*****************************************************************/
        SRMODE = lo;                      ! Set Status Register To User Mode
        D[1] = 0x55555555;                ! Place Hex 55555555 Into D[1]
        D[2] = 0x00000000;                ! Place 0 Into D[2]
        A[0] = 0x1000;                    ! Place Hex 1000 Into A[0]
        PC = 0x1000;                      ! Place Hex 1000 Into Program Counter
        next                              ! Execute Assignments
        )

/*************************************************************************/
/*                                                                       */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary    */
/*  to retrieve modeled instructions for simulation and analysis. It    */
/*  was foshsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                     */
/*                                                                     */
/*********************************************************************/

fetch_initial_instruction :=
      (

      /****************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /****************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 2
      while DTACKN eql hi                     ! Wait For Memory To Place
```

```
          (                                    ! Data On The Bus
          next;                                ! Execute Impending Assignments

          PHI1 = lo;                           ! Phase 2
          PHI2 = hi;                           ! Of Clock Cycle 2
          next;                                ! Execute Assignments

          /****************************************************************/
          T = 3;                               ! Clock Cycle 3
          next;                                ! Execute Assignment

          PHI1 = hi;                           ! Phase 1
          PHI2 = lo;                           ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
          DTACKN = lo;                         ! Asserts DTACKN(Added)
          next;                                ! Execute Pending Assignments

          /****************************************************************/
          T = 2                                ! Return To  Phase 2
                                               ! Of Clock Cycle 2
          );
          next;                                ! Execute Impending Assignments

     /**********************************************************************/
     T = 3;                                    ! Clock Cycle 3
     next;                                     ! Execute Assignment

     PHI1 = lo;                                ! Phase 2
     PHI2 = hi;                                ! Of Clock Cycle 3
     EXDBUF = DBUS;                            ! Instruction On Data Bus
                                               ! Is Placed In External Data
                                               ! Bus Buffer
     next;                                     ! Execute Pending Assignments

     /**********************************************************************/
     T = 4;                                    ! Clock Cycle 4
     next;                                     ! Execute Assignment

     PHI1 = hi;                                ! Phase 1
     PHI2 = lo;                                ! Of Clock Cycle 4
     PFR = EXDBUF;                             ! The Contents Of The External
                                               ! Data Bus Buffer Are Placed
                                               ! In Prefetch Register
     next;                                     ! Execute Pending Assignments

     PHI1 = lo;                                ! Phase 2
     PHI2 = hi;                                ! Of Clock Cycle 4
     ASN = hi;                                 ! Deactivate Address Strobe
     LDSN = hi;                                ! Deactivate Lower Data Strobe
     UDSN = hi;                                ! Deactivate Upper Data Strobe
     IR = PFR;                                 ! Contents Of Prefetch Register
                                               ! Are Placed Into Instruction
```

```
                                            ! Register
        DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                            ! Acknowledge
        PC = PC + 2;                        ! Increment Program Counter
        next;                               ! Execute Pending Assignments
        T = 0                               ! Reset Clock Cycle Counter
        )

  branch :=
         (
         next;

        /**************************************************************/
         T = 9;                             ! Clock Cycle 9
         next;

         PHI1 = hi;
         PHI2 = lo;
         PC = PC + IR<7:0>;                 ! Add Branch Displacement To PC
         ADENABLE = lo;                     ! Disable Address Bus
         DBENABLE = lo;                     ! Disable Data Bus
         DBUS = 0xffff;                     ! Data Bus High Impedanced
         next;

         PHI1 = lo;
         PHI2 = hi;
         next;

        /**************************************************************/
         T = 10;                            ! Clock Cycle 10
         next;

         PHI1 = hi;
         PHI2 = lo;
         next;

         PHI1 = lo;
         PHI2 = hi;
         next;

        /**************************************************************/
         T = 11;                            ! Clock Cycle 11
         next;

        PHI1 = hi;                          ! Phase 1 Of
        PHI2 = lo;                          ! Clock Cycle 11
        RW = hi;                            ! Memory Read
        ADENABLE = lo;                      ! Disable Address Bus Buffer
        DBENABLE = lo;                      ! Disable Data Bus Buffer
        DBUS = 0xffff;                      ! Data Bus In High Impedance
        IABUS = PC;                         ! Place PC On Internal Address
                                            ! Bus
        next;                               ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 11
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 12;                                 ! Clock Cycle 12
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 12
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 12
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 13;                                 ! Clock Cycle 13
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 13
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 13
      next;                             ! Execute Assignments

      /**********************************************************/
      T = 14;                           ! Clock Cycle 14
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 14
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments
```

```
        /************************************************************/
        T = 13                              ! Return To  Phase 2
                                            ! Of Clock Cycle 13
        );
        next;                               ! Execute Impending Assignments

    /****************************************************************/
    T = 14;                                 ! Clock Cycle 14
    next;                                   ! Execute Assignment

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 14
    EXDBUF = DBUS;                          ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
    next;                                   ! Execute Pending Assignments

    /****************************************************************/
    T = 15;                                 ! Clock Cycle 15
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 15
    PFR = EXDBUF;                           ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
    next;                                   ! Execute Pending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 15
    ASN = hi;                               ! Deactivate Address Strobe
    IR = PFR;                               ! Place Prefetch Register Into IR
    DTACKN = hi;                           ! Activate Data Transfer Acknowledge
    LDSN = hi;                              ! Deactivate Lower Data Strobe
    UDSN = hi;                              ! Deactivate Upper Data Strobe
    PC = PC + 2;                            ! Increment Program Counter
    next;                                   ! Execute Pending Assignments

    /****************************************************************/
    T = 0                                   ! Reset Clock Cycle Counter
    )

nobranch :=
      (
      PC = PC + 2;                          ! Incrment PC
      next;

    /****************************************************************/
      T = 5;                            ! Clock Cycle 5
      next;

      PHI1 = hi;
      PHI2 = lo;
```

```
           ADENABLE = lo;                          ! Disable Address Bus
           DBENABLE = lo;                          ! Disable Data bus
           DBUS = 0xffff;                          ! Data Bus High Impedanced
           next;

           PHI1 = lo;
           PHI2 = hi;
           next;

        /***************************************************************/
           T = 6;                              ! Clock Cycle 6
           next;

           PHI1 = hi;
           PHI2 = lo;
           next;

           PHI1 = lo;
           PHI2 = hi;
           next;

        /***************************************************************/
           T = 7;                              ! Clock Cycle 7
           next;

           PHI1 = hi;
           PHI2 = lo;
           next;

           PHI1 = lo;
           PHI2 = hi;
           next;

        /***************************************************************/
           T = 8;                              ! Clock Cycle 8
           next;

           PHI1 = hi;
           PHI2 = lo;
           next;

           PHI1 = lo;
           PHI2 = hi;
           IR = PFR;                           ! Place Prefetch Register Into IR
           next;

        /***************************************************************/
           T = 0;                              ! Reset Clock Cycle Counter
           )

beq     :=                                         ! BEQ $1000
        (
/*****************************************************************/

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 0
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus
DBENABLE = lo;                          ! Disable Data Bus
DBUS = 0xffff;                           ! Data Bus In High Impedance State
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 0
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments
```

```
          /*************************************************************/
          T = 3;                                  ! Clock Cycle 3
          next;                                   ! Execute Assignment

          PHI1 = hi;                              ! Phase 1
          PHI2 = lo;                              ! Of Clock Cycle 3
          DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
          DTACKN = lo;                            ! Asserts DTACKN(Added)
          next;                                   ! Execute Pending Assignments

          /*************************************************************/
          T = 2                                   ! Return To  Phase 2
                                                  ! Of Clock Cycle 2
          );
          next;                                   ! Execute Impending Assignments

      /*****************************************************************/
      T = 3;                                      ! Clock Cycle 3
      next;                                       ! Execute Assignment

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 3
      EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
      next;                                       ! Execute Pending Assignments

      /*****************************************************************/
      T = 4;                                      ! Clock Cycle 4
      next;                                       ! Execute Assignment

      PHI1 = hi;                                  ! Phase 1
      PHI2 = lo;                                  ! Of Clock Cycle 4
      PFR = EXDBUF;                               ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Prefetch Register
      next;                                       ! Execute Pending Assignments

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 4
      DTACKN = hi;                                ! Deactivate Data Transfer(Added)
      ASN = hi;                                 ! Deactivate Address Strobe
      LDSN = hi;                                ! Deactivate Lower Data Strobe
      UDSN = hi;                                ! Deactivate Upper Data Strobe
      next;                                       ! Acknowledge
      if SRZERO                                   ! If Zero Bit Is Set Then
         branch                                   ! Execute Branch Cycle
      else nobranch                               ! Else Execute Nonbranch Cycle
      )

move :=                                           ! MOVE.W D1,D3 and MOVE.W D2,D3
```

```
(

/*****************************************************************/

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 0
DBUS = 0xffff;                            ! Place Data Bus In High Impedance
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS = PC;                               ! Place PC On Internal Address
                                          ! Bus
case IR                                   ! Place Low Word From D[1]/D[2] Onto
     0x3601: IDBUS = D1LWORD              ! Internal Data Bus
     0x3602: IDBUS = D2LWORD
esac;
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 0
ADENABLE = hi;                            ! Enable Address Bus Buffer
EXABUF = IABUS;                           ! Gate Internal Address Bus
                                          ! Into External Address Buffer
FCMODE = SRMODE;                          ! User Mode
FCSPACE = 2;                              ! Accessing Program
SRCARRY = lo;                             ! Clear Status Register Carry Bit
SROVER = lo;                              ! Clear Status Register Overflow Bit
SRZERO = lo;                              ! Clear Status Register Zero Bit
SRNEG  = lo;                              ! Clear Status Register Negative Bit
D3LWORD = IDBUS;                          ! Place Data From Internal Data Bus
                                          ! Into Low Word Of D[3]
next;                                     ! Execute Impending Assignments
ABUS = EXABUF;                            ! Address Placed On Bus(Added)
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 1;                                    ! Clock Cycle 1
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 1
ASN = lo;                                 ! Assert Address Strobe
LDSN = lo;                                ! Assert Lower Data Strobe
UDSN = lo;                                ! Assert Upper Data Strobe
DBENABLE = hi;                            ! Enable Data Bus
if D3LWORD eql 0                          ! Set Status Register Zero Bit
   SRZERO = hi;                           ! If Moved Data Is Zero
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 1
if D[2]<15>                               ! Set Status Register Negative
   SRNEG = hi;                            ! Bit If Moved Data Is Negative
```

```
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***********************************************************/
      T = 2                             ! Return To  Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
```

```
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )


jmp :=                                          ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
```

```
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/*****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
     (                                  ! Data On The Bus
     next;                              ! Execute Impending Assignments

     PHI1 = lo;                         ! Phase 2
     PHI2 = hi;                         ! Of Clock Cycle 2
     next;                              ! Execute Assignments

     /***********************************************************/
     T = 3;                             ! Clock Cycle 3
     next;                              ! Execute Assignment

     PHI1 = hi;                         ! Phase 1
     PHI2 = lo;                         ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
     DTACKN = lo;                       ! Asserts DTACKN(Added)
     next;                              ! Execute Pending Assignments

     /***********************************************************/
     T = 2                              ! Return To Phase 2
                                        ! Of Clock Cycle 2
     );
     next;                              ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
```

```
DBENABLE = hi;                                  ! Enable Data Bus
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 6
next;                                           ! Execute Pending Assignments

/****************************************************************************/
T = 7;                                          ! Clock Cycle 7
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 7
while DTACKN eql hi                             ! Wait For Memory To Place
     (                                          ! Data On The Bus
     next;                                      ! Execute Impending Assignments

     PHI1 = lo;                                 ! Phase 2
     PHI2 = hi;                                 ! Of Clock Cycle 7
     next;                                      ! Execute Assignments

     /**********************************************************************/
     T = 8;                                     ! Clock Cycle 8
     next;                                      ! Execute Assignment

     PHI1 = hi;                                 ! Phase 1
     PHI2 = lo;                                 ! Of Clock Cycle 8
     DBUS<15:8> = M[ABUS];                      ! Memory Places Instruction
     DBUS<7:0> = M[ABUS + 1];                   ! On Data Bus And
     DTACKN = lo;                               ! Asserts DTACKN(Added)
     next;                                      ! Execute Pending Assignments

     /**********************************************************************/
     T = 7                                    ! Return To Phase 2
                                              ! Of Clock Cycle 7
     );
     next;                                    ! Execute Impending Assignments

/***************************************************************************/
T = 8;                                          ! Clock Cycle 8
next;                                           ! Execute Assignment

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 8
EXDBUF = DBUS;                                  ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
next;                                           ! Execute Pending Assignments

/***************************************************************************/
T = 9;                                          ! Clock Cycle 9
next;                                           ! Execute Assignment
```

```
        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                    (
                    case IR
                         0x67fc,0x67f8: beq   ! BEQ $1000
                         0x3601,0x3602: move  ! MOVE.W D1,D3 and D2,D3
                                0x4ed0: jmp   ! JMP (A0) If IR = Octal Value
                    esac
                    )

    main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
            (
            decode_execute_prefetch
            )
        )
```

```
/*************************************************************************/
/*                                                                       */
/*     MOTOROLA MC68000 MODEL OF THE BTST D1,(A1) INSTRUCTION            */
/*                                                                       */
/*************************************************************************/

/*************************************************************************/
/*                                                                       */
/*                       Structure Declarations                          */
/*                                                                       */
/*************************************************************************/

state

/*************************************************************************/
/*                                                                       */
/*                  M68000 Programming Registers                         */
/*                                                                       */
/*************************************************************************/

D[0:7]<31:0>,                  ! 8 Data Registers
A[0:6]<31:0>,                  ! 7 Address Registers
UA7<31:0>,                     ! User Stack Pointer
SA7<31:0>,                     ! System Stack Pointer
PC<31:0>,                      ! Program Counter
SR<15:0>,                      ! Status Register

/*************************************************************************/
/*                                                                       */
/*                  Temporary Internal Registers                         */
/*                                                                       */
/*************************************************************************/

PFR<15:0>,                     ! Prefetch Register
IR<15:0>,                      ! Instruction Register
FC<2:0>,                       ! Function Code Register
EXDBUF<15:0>,                  ! External Data Bus Buffer Register
EXABUF<23:1>,                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,                 ! ALU Buffer 1
ALUBUF2<31:0>,                 ! ALU Buffer 2
DTEMP<15:0>,                   ! Temporary Data Storage
DISREG<31:0>,                  ! Temporary Displacement Storage
SRTEMP<15:0>,                  ! Temporary Status Register Storage
                               ! (Exception Processing)
IRTEMP<15:0>,                  ! Temporary Instruction Register Storage
                               ! (Exception Processing)
TEMPADR<31:0>,                 ! Temporary Cycle Address Storage
                               ! (Exception Processing)
ACTYPE<15:0>,                  ! Temporary Access Type Storage
                               ! (Exception Processing)
VECADR<23:0>,                  ! Temporary Vector Address Storage
                               ! (Exception Processing)
```

```
HANADR<31:0>,              ! Temporary Address Storage For
                           ! Exception Handler Routine
T<7:0>,                    ! Clock Cycle Counter
HOLD<3:0>,                 ! Temorary Holding Register
RESET,                   ! Reset Flip-Flop
HALT,                    ! Halt Flip-Flop
RW,                      ! Read/Write Flip-Flop
ADENABLE,                ! Address Bus Buffer Enable
DBENABLE,                ! Data Bus Buffer Enable
ASN,                     ! Address Strobe Flip-Flop
LDSN,                    ! Lower Data Strobe Flip-Flop
UDSN,                    ! Upper Data Strobe Flip-Flop
DTACKN,                  ! Data Transfer Acknowledge Flip-Flop
COUT,                    ! Carry Flip-Flop
EXCEPT,                  ! Exception Processing Flip-Flop
READY,                   ! Ready Flip-Flop

/**********************************************************************/
/*                                                                    */
/*      Model transformation modifications:                          */
/*                                                                    */
/*            1) CDL decoder structure nonexistent in ISP' and un-   */
/*      necessary for model. Eliminated.                             */
/*            2) Multi-phase clock structure nonexistent in ISP'.    */
/*      Operations on registers will provide its equivalent.         */
/*            3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                        */
/*            4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                  */
/*            5) The memory word length was reduced from 16 to 8 bit */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing   */
/*      MC68000 assembler and linker/loader models. The memory was   */
/*      also reduced from 8 Mwords to 32 Kbytes.                     */
/*                                                                    */
/**********************************************************************/

IABUS<31:0>,               ! Internal Address Bus
IDBUS<31:0>,               ! Internal Data Bus
SWITCH,                  ! Power Switch
PHI1,                    ! Phase 1 Of Two-Phase Clock
PHI2;                    ! Phase 2 Of Two-Phase Clock

port

/**********************************************************************/
/*                                                                    */
/*                External Address and Data Bus                      */
/*                                                                    */
/**********************************************************************/

DBUS<15:0>,                ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/*************************************************************************/
/*                                                                       */
/*                        Register Subfields                             */
/*                                                                       */
/*************************************************************************/

PCADDR       = PC<23:0>;      ! Program Counter Address Field
SRTRACE      = SR<15>,        ! Trace Bit
SRMODE       = SR<13>,        ! Mode Selection Bit
SRCARRY      = SR<0>,         ! Carry Bit
SROVER       = SR<1>,         ! Overflow Bit
SRZERO       = SR<2>,         ! Zero Bit
SRNEG        = SR<3>,         ! Negative Bit
SREX         = SR<4>,         ! Extend Bit
SRMASK       = SR<10:8>,      ! Interrupt Mask
FCSPACE      = FC<1:0>,       ! Memory Access Address Space
FCMODE       = FC<2>,         ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,      ! PC Low Word
PCHI         = PC<31:16>,     ! PC High Word
D0LWORD      = D[0]<15:0>,    ! D[0] Low Word
D1LWORD      = D[1]<15:0>,    ! D[1] Low Word
D2LWORD      = D[2]<15:0>,    ! D[2] Low Word
D3LWORD      = D[3]<15:0>,    ! D[3] Low Word
D4LWORD      = D[4]<15:0>,    ! D[4] Low Word
D5LWORD      = D[5]<15:0>,    ! D[5] Low Word
D6LWORD      = D[6]<15:0>,    ! D[6] Low Word
D7LWORD      = D[7]<15:0>,    ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/*************************************************************************/
/*                                                                       */
/*                    16K 16-Bit Word Internal Memory                    */
/*                                                                       */
/*************************************************************************/

M[0:32767]<7:0>;

macro

/*************************************************************************/
/*                                                                       */
/*                       Logic Level Macros                              */
/*                                                                       */
/*************************************************************************/

lo    = 0 %,
hi    = 1 %,
off   = 0 %,
on    = 1 %,
clear = 0 %;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                        ! Turn Power On
        next;                               ! Execute Assignment
        READY = lo;                         ! System Not Ready
        RESET = lo;                         ! Assert Reset For
        delay(100);                         ! 100 Miliseconds(Active Low)
        RESET = hi;                         ! Deactivate Reset
        next;                               ! Execute Pending Assignments
        ASN = hi;                           ! Initialize Address Strobe
        LDSN = hi;                          ! Initialize Lower Data Strobe
        UDSN = hi;                          ! Initialize Upper Data Strobe
        DTACKN = hi;                        ! Initialize Data Transfer Acknowledge
        RW = hi;                            ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                      ! Place Data Bus In High Impedance State
        M[0x1008] = 0xff;                   ! Place Memory Locations Following The
        M[0x1009] = 0xff;                   ! JMP Instruction In A High State
        HALT = hi;                          ! Initialize Halt Flip-Flop(Active
                                            ! Low)
        T = 0;                              ! Initialize Clock Cycle Counter
        READY = hi;                         ! System Ready
        /*********************************************************************/
        /*                                                                   */
        /*      Routine Initialization Per Hamby and Guillory                */
        /*                                                                   */
        /*********************************************************************/
        SRMODE = lo;                        ! Set Status Register To User Mode
        D[1] = 0x3;                         ! Place Hex 3 Into D[1]
        D[2] = 0x55555555;                  ! Place Hex 55555555 Into D[2]
        A[0] = 0x1000;                      ! Place Hex 1000 Into A[0]
        A[1] = 0x2001;                      ! Place Hex 2001 Into A[1]
        M[0x2000] = 0x0;                    ! Initialize Location 2000 To Zero
        M[0x2001] = 0x55;                   ! Initialize Location 2001 To Hex 55
        PC = 0x1000;                        ! Place Hex 1000 Into Program Counter
        next                                ! Execute Assignments
        )
```

```
/**************************************************************************/
/*                                                                        */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary     */
/*  to retrieve modeled instructions for simulation and analysis. It     */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory  */
/*  on page VI-15 of their thesis.                                       */
/*                                                                        */
/**************************************************************************/

fetch_initial_instruction :=
      (

      /*******************************************************************/

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 0
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = PC;                             ! Place PC On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 0
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      EXABUF = IABUS;                         ! Gate Internal Address Bus
                                              ! Into External Address Buffer
      FCMODE = SRMODE;                        ! User Mode
      FCSPACE = 2;                            ! Accessing Program
      next;                                   ! Execute Impending Assignments
      ABUS = EXABUF;                          ! Address Placed On Bus(Added)
      next;                                   ! Execute Pending Assignments

      /*******************************************************************/
      T = 1;                                  ! Clock Cycle 1
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 1
      ASN = lo;                               ! Assert Address Strobe
      LDSN = lo;                              ! Assert Lower Data Strobe
      UDSN = lo;                              ! Assert Upper Data Strobe
      DBENABLE = hi;                          ! Enable Data Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 1
      next;                                   ! Execute Pending Assignments

      /*******************************************************************/
      T = 2;                                  ! Clock Cycle 2
      next;                                   ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 2
    next;                               ! Execute Assignments

    /***************************************************************/
    T = 3;                              ! Clock Cycle 3
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 3
    DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
    DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
    DTACKN = lo;                        ! Asserts DTACKN(Added)
    next;                               ! Execute Pending Assignments

    /***************************************************************/
    T = 2                               ! Return To  Phase 2
                                        ! Of Clock Cycle 2
    );
    next;                               ! Execute Impending Assignments

/*******************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 3
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*******************************************************************/
T = 4;                                  ! Clock Cycle 4
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 4
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
```

```
        LDSN = hi;                          ! Deactivate Lower Data Strobe
        UDSN = hi;                          ! Deactivate Upper Data Strobe
        IR = PFR;                           ! Contents Of Prefetch Register
                                            ! Are Placed Into Instruction
                                            ! Register
        DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                            ! Acknowledge
        PC = PC + 2;                        ! Increment Program Counter
        next;                               ! Execute Pending Assignments
        T = 0                               ! Reset Clock Cycle Counter
        )

btst    :=                                  ! BTST D1,(A1)
        (

        /****************************************************************/

        PHI1 = hi;                          ! Phase 1 Of
        PHI2 = lo;                          ! Clock Cycle 0
        RW = hi;                            ! Memory Read
        ADENABLE = lo;                      ! Disable Address Bus Buffer
        DBUS = 0xffff;                      ! Data Bus High Impedanced
        DBENABLE = lo;                      ! Disable Data Bus Buffer
        IABUS = PC;                         ! Place PC On Internal Address
                                            ! Bus
        next;                               ! Execute Pending Assignments

        PHI1 = lo;                          ! Phase 2 Of
        PHI2 = hi;                          ! Clock Cycle 0
        ADENABLE = hi;                      ! Enable Address Bus Buffer
        EXABUF = IABUS;                     ! Gate Internal Address Bus
                                            ! Into External Address Buffer
        FCMODE = SRMODE;                    ! User Mode
        FCSPACE = 2;                        ! Accessing Program
        next;                               ! Execute Impending Assignments
        ABUS = EXABUF;                      ! Address Placed On Bus(Added)
        next;                               ! Execute Pending Assignments

        /****************************************************************/
        T = 1;                              ! Clock Cycle 1
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1 Of
        PHI2 = lo;                          ! Clock Cycle 1
        ASN = lo;                           ! Assert Address Strobe
        LDSN = lo;                          ! Assert Lower Data Strobe
        UDSN = lo;                          ! Assert Upper Data Strobe
        DBENABLE = hi;                      ! Enable Data Bus
        next;                               ! Execute Pending Assignments

        PHI1 = lo;                          ! Phase 2
        PHI2 = hi;                          ! Of Clock Cycle 1
        next;                               ! Execute Pending Assignments
```

```
/*****************************************************************/
T = 2;                                      ! Clock Cycle 2
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 2
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 2
      next;                                 ! Execute Assignments

      /**********************************************************/
      T = 3;                                ! Clock Cycle 3
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /**********************************************************/
      T = 2                                 ! Return To  Phase 2
                                            ! Of Clock Cycle 2
      );
      next;                                 ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                      ! Clock Cycle 4
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 4
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
                                        ! Register
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
PC = PC + 2;                            ! Increment Program Counter
next;                                   ! Execute Pending Assignments

/******************************************************************/

T = 5;
next;

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
DBUS = 0xffff;                          ! Data Bus High Impedanced
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Program
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/******************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments
```

```
/*************************************************************************/
T = 7;                                    ! Clock Cycle 7
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 7
while DTACKN eql hi                       ! Wait For Memory To Place
    (                                     ! Data On The Bus
    next;                                 ! Execute Impending Assignments

    PHI1 = lo;                            ! Phase 2
    PHI2 = hi;                            ! Of Clock Cycle 7
    next;                                 ! Execute Assignments

    /********************************************************************/
    T = 8;                                ! Clock Cycle 8
    next;                                 ! Execute Assignment

    PHI1 = hi;                            ! Phase 1
    PHI2 = lo;                            ! Of Clock Cycle 8
    DBUS<7:0> = M[ABUS];                  ! Place Byte On Data Bus And
    DTACKN = lo;                          ! Asserts DTACKN(Added)
    next;                                 ! Execute Pending Assignments

    /********************************************************************/
    T = 7                                 ! Return To  Phase 2
                                          ! Of Clock Cycle 7
    );
    next;                                 ! Execute Impending Assignments

/*************************************************************************/
T = 8;                                    ! Clock Cycle 8
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 8
EXDBUF = DBUS;                            ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
next;                                     ! Execute Pending Assignments

/*************************************************************************/
T = 9;                                    ! Clock Cycle 9
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 9
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 9
HOLD = D[1]<2:0> ext 4;                   ! Place First 3 Bits Of D[1]
                                          ! Into Temporary Register
```

```
        case HOLD                                      ! The First 3 Bits Of D[1]
             0: if EXDBUF<0>
                   SRZERO = hi
                else
                   SRZERO = lo
             1: if EXDBUF<1>
                   SRZERO = hi
                else
                   SRZERO = lo
             2: if EXDBUF<2>                           ! Determine Which Bit Of
                   SRZERO = hi
                else
                   SRZERO = lo
             3: if EXDBUF<3>
                   SRZERO = hi
                else
                   SRZERO = lo
             4: if EXDBUF<4>                           ! Data Will Be Checked For Set
                   SRZERO = hi
                else
                   SRZERO = lo
             5: if EXDBUF<5>
                   SRZERO = hi
                else
                   SRZERO = lo
             6: if EXDBUF<6>
                   SRZERO = hi
                else
                   SRZERO = lo
             7: if EXDBUF<7>
                   SRZERO = hi
                else
                   SRZERO = lo
        esac;
        ASN = hi;                                       ! Deactivate Address Strobe
        LDSN = hi;                                      ! Deactivate Lower Data Strobe
        IR = PFR;                                       ! Contents Of Prefetch Register
                                                        ! Are Placed Into Instruction
                                                        ! Register
        DTACKN = hi;                                    ! Deactivate Data Transfer(Added)
                                                        ! Acknowledge
        next;                                           ! Execute Pending Assignments
        T = 0                                           ! Reset Clock Cycle Counter
        )

move :=                                                 ! MOVE.W D2,D3
        (

        /*****************************************************************/

        PHI1 = hi;                                      ! Phase 1 Of
        PHI2 = lo;                                      ! Clock Cycle 0
        DBUS = 0xffff;                                  ! Place Data Bus In High Impedance
```

```
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
IDBUS = D2LWORD;                        ! Place Low Word From D[2] Onto
                                        ! Internal Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 0
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
SRCARRY = lo;                           ! Clear Status Register Carry Bit
SROVER = lo;                            ! Clear Status Register Overflow Bit
SRZERO = lo;                            ! Clear Status Register Zero Bit
SRNEG  = lo;                            ! Clear Status Register Negative Bit
D3LWORD = IDBUS;                        ! Place Data From Internal Data Bus
                                        ! Into Low Word Of D[3]
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 1;                                  ! Clock Cycle 1
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
if D3LWORD eql 0                        ! Set Status Register Zero Bit
   SRZERO = hi;                         ! If Moved Data Is Zero
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
if D[3]<15>                             ! Set Status Register Negative
   SRNEG = hi;                          ! Bit If Moved Data Is Negative
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
```

```
        (                                    ! Data On The Bus
        next;                                ! Execute Impending Assignments

        PHI1 = lo;                           ! Phase 2
        PHI2 = hi;                           ! Of Clock Cycle 2
        next;                                ! Execute Assignments

        /*********************************************************/
        T = 3;                               ! Clock Cycle 3
        next;                                ! Execute Assignment

        PHI1 = hi;                           ! Phase 1
        PHI2 = lo;                           ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
        DTACKN = lo;                         ! Asserts DTACKN(Added)
        next;                                ! Execute Pending Assignments

        /*********************************************************/
        T = 2                                ! Return To  Phase 2
                                             ! Of Clock Cycle 2
        );
        next;                                ! Execute Impending Assignments

    /*************************************************************/
    T = 3;                                   ! Clock Cycle 3
    next;                                    ! Execute Assignment

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 3
    EXDBUF = DBUS;                           ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
    next;                                    ! Execute Pending Assignments

    /*************************************************************/
    T = 4;                                   ! Clock Cycle 4
    next;                                    ! Execute Assignment

    PHI1 = hi;                               ! Phase 1
    PHI2 = lo;                               ! Of Clock Cycle 4
    PFR = EXDBUF;                            ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
    next;                                    ! Execute Pending Assignments

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 4
    ASN = hi;                                ! Deactivate Address Strobe
    LDSN = hi;                               ! Deactivate Lower Data Strobe
    UDSN = hi;                               ! Deactivate Upper Data Strobe
    IR = PFR;                                ! Contents Of Prefetch Register
                                             ! Are Placed Into Instruction
```

```
                                        ! Register
        DTACKN = hi;                    ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
        PC = PC + 2;                    ! Increment Program Counter
        next;                           ! Execute Impending Assignments
        T = 0                           ! Reset Clock Cycle Counter
        )


    jmp :=                              ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                      ! Phase 1 Of
        PHI2 = lo;                      ! Clock Cycle 0
        DBUS = 0xffff;                  ! Place Data Bus In A High Impedance
        RW = hi;                        ! Memory Read
        ADENABLE = lo;                  ! Disable Address Bus Buffer
        DBENABLE = lo;                  ! Disable Data Bus Buffer
        IABUS = PC;                     ! Place PC On Internal Address
                                        ! Bus
        next;                           ! Execute Pending Assignments

        PHI1 = lo;                      ! Phase 2 Of
        PHI2 = hi;                      ! Clock Cycle 0
        ADENABLE = hi;                  ! Enable Address Bus Buffer
        EXABUF = IABUS;                 ! Gate Internal Address Bus
                                        ! Into External Address Buffer
        FCMODE = SRMODE;                ! User Mode
        FCSPACE = 2;                    ! Accessing Program
        next;                           ! Execute Pending Assignments
        ABUS = EXABUF;                  ! Address Placed On Bus(Added)
        next;                           ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                          ! Clock Cycle 1
        next;                           ! Execute Assignment

        PHI1 = hi;                      ! Phase 1 Of
        PHI2 = lo;                      ! Clock Cycle 1
        ASN = lo;                       ! Assert Address Strobe
        LDSN = lo;                      ! Assert Lower Data Strobe
        UDSN = lo;                      ! Assert Upper Data Strobe
        IABUS = A[0];                   ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
        DBENABLE = hi;                  ! Enable Data Bus
        next;                           ! Execute Pending Assignments

        PHI1 = lo;                      ! Phase 2
        PHI2 = hi;                      ! Of Clock Cycle 1
        PC = IABUS;                     ! Place Jump Address Into Program
```

```
                                                ! Counter
     next;

     /*****************************************************************/
     T = 2;                                     ! Clock Cycle 2
     next;                                      ! Execute Assignment

     PHI1 = hi;                                 ! Phase 1
     PHI2 = lo;                                 ! Of Clock Cycle 2
     while DTACKN eql hi                        ! Wait For Memory To Place
           (                                    ! Data On The Bus
           next;                                ! Execute Impending Assignments

           PHI1 = lo;                           ! Phase 2
           PHI2 = hi;                           ! Of Clock Cycle 2
           next;                                ! Execute Assignments

           /***********************************************************/
           T = 3;                               ! Clock Cycle 3
           next;                                ! Execute Assignment

           PHI1 = hi;                           ! Phase 1
           PHI2 = lo;                           ! Of Clock Cycle 3
           DBUS<15:8> = M[ABUS];                ! Memory Places Instruction
           DBUS<7:0> = M[ABUS + 1];             ! On Data Bus And
           DTACKN = lo;                         ! Asserts DTACKN(Added)
           next;                                ! Execute Pending Assignments

           /***********************************************************/
           T = 2                                ! Return To Phase 2
                                                ! Of Clock Cycle 2
           );
           next;                                ! Execute Impending Assignments

     /*****************************************************************/
     T = 3;                                     ! Clock Cycle 3
     next;                                      ! Execute Assignment

     PHI1 = lo;                                 ! Phase 2
     PHI2 = hi;                                 ! Of Clock Cycle 3
     EXDBUF = DBUS;                             ! Instruction On Data Bus
                                                ! Is Placed In External Data
                                                ! Bus Buffer
     next;                                      ! Execute Pending Assignments

     /*****************************************************************/
     T = 4;                                     ! Clock Cycle 4
     next;                                      ! Execute Assignment

     PHI1 = hi;                                 ! Phase 1
     PHI2 = lo;                                 ! Of Clock Cycle 4
     next;
     PFR = EXDBUF;                              ! The Contents Of The External
```

```
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/**************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/**************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/**************************************************************/
T = 7;                                  ! Clock Cycle 7
```

```
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 7
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 7
      next;                             ! Execute Assignments

      /***************************************************************/
      T = 8;                            ! Clock Cycle 8
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /***************************************************************/
      T = 7                            ! Return To Phase 2
                                       ! Of Clock Cycle 7
      );
      next;                            ! Execute Impending Assignments

/*********************************************************************/
T = 8;                                  ! Clock Cycle 8
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 8
EXDBUF = DBUS;                          ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
next;                                   ! Execute Pending Assignments

/*********************************************************************/
T = 9;                                  ! Clock Cycle 9
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 9
PFR = EXDBUF;                           ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 9
```

```
        ASN = hi;                                   ! Deactivate Address Strobe
        LDSN = hi;                                  ! Deactivate Lower Data Strobe
        UDSN = hi;                                  ! Deactivate Upper Data Strobe
        PC = PC + 2;                                ! Increment Program Counter
        IR = PFR;                                   ! Place Contents Of Prefetch
                                                    ! Register Into Instruction
                                                    ! Register
        DTACKN = hi;                                ! Deactivate Data Transfer
                                                    ! Acknowledge(Added)
        next;                                       ! Execute Pending Assignments
        T = 0                                       ! Reset Clock Cycle Counter
        )

decode_execute_prefetch :=
                            (
                            case IR
                                 0x0311: btst       ! BTST D1,(A1)
                                 0x3602: move       ! MOVE.W D2,D3
                                 0x4ed0: jmp        ! JMP (A0) If IR = Octal Value
                            esac
                            )

main :=
      (
      power_on_initialize;
      fetch_initial_instruction;
      while READY eql hi
            (
            decode_execute_prefetch
            )
      )
```

```
/**********************************************************************/
/*                                                                    */
/*     MOTOROLA MC68000 MODEL OF THE ILLEGAL INSTRUCTION              */
/*                                                                    */
/**********************************************************************/

/**********************************************************************/
/*                                                                    */
/*                   Structure Declarations                           */
/*                                                                    */
/**********************************************************************/

state

/**********************************************************************/
/*                                                                    */
/*               M68000 Programming Registers                         */
/*                                                                    */
/**********************************************************************/

D[0:7]<31:0>,               ! 8 Data Registers
A[0:6]<31:0>,               ! 7 Address Registers
UA7<31:0>,                  ! User Stack Pointer
SA7<31:0>,                  ! System Stack Pointer
PC<31:0>,                   ! Program Counter
SR<15:0>,                   ! Status Register

/**********************************************************************/
/*                                                                    */
/*               Temporary Internal Registers                         */
/*                                                                    */
/**********************************************************************/

PFR<15:0>,                  ! Prefetch Register
IR<15:0>,                   ! Instruction Register
FC<2:0>,                    ! Function Code Register
EXDBUF<15:0>,               ! External Data Bus Buffer Register
EXABUF<23:1>,               ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,              ! ALU Buffer 1
ALUBUF2<31:0>,              ! ALU Buffer 2
DTEMP<15:0>,                ! Temporary Data Storage
DISREG<31:0>,               ! Temporary Displacement Storage
SRTEMP<15:0>,               ! Temporary Status Register Storage
                            ! (Exception Processing)
IRTEMP<15:0>,               ! Temporary Instruction Register Storage
                            ! (Exception Processing)
TEMPADR<31:0>,              ! Temporary Cycle Address Storage
                            ! (Exception Processing)
ACTYPE<15:0>,               ! Temporary Access Type Storage
                            ! (Exception Processing)
VECADR<23:0>,               ! Temporary Vector Address Storage
                            ! (Exception Processing)
```

```
HANADR<31:0>,               ! Temporary Address Storage For
                            ! Exception Handler Routine
T<7:0>,                     ! Clock Cycle Counter
RESET,                   ! Reset Flip-Flop
HALT,                    ! Halt Flip-Flop
RW,                      ! Read/Write Flip-Flop
ADENABLE,                ! Address Bus Buffer Enable
DBENABLE,                ! Data Bus Buffer Enable
ASN,                     ! Address Strobe Flip-Flop
LDSN,                    ! Lower Data Strobe Flip-Flop
UDSN,                    ! Upper Data Strobe Flip-Flop
DTACKN,                  ! Data Transfer AcKnowledge Flip-Flop
COUT,                    ! Carry Flip-Flop
EXCEPT,                  ! Exception Processing Flip-Flop
READY,                   ! Ready Flip-Flop

/*********************************************************************/
/*                                                                  */
/*      Model transformation modifications:                         */
/*                                                                  */
/*          1) CDL decoder structure nonexistent in ISP' and un-    */
/*      necessary for model. Eliminated.                            */
/*          2) Multi-phase clock structure nonexistent in ISP'.     */
/*      Operations on registers will provide its equivalent.        */
/*          3) Switch structure nonexistent in ISP'. Operation on a */
/*      register will provide its equivalent.                       */
/*          4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model */
/*      equivalency and simplicity.                                 */
/*          5) The memory word length was reduced from 16 to 8 bit  */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing  */
/*      MC68000 assembler and linker/loader models. The memory was  */
/*      also reduced from 8 Mwords to 32 Kbytes.                    */
/*                                                                  */
/*********************************************************************/

IABUS<31:0>,                ! Internal Address Bus
IDBUS<31:0>,                ! Internal Data Bus
twait<7:0>,
SWITCH,                  ! Power Switch
PHI1,                    ! Phase 1 Of Two-Phase Clock
PHI2;                    ! Phase 2 Of Two-Phase Clock

port

/*********************************************************************/
/*                                                                  */
/*              External Address and Data Bus                       */
/*                                                                  */
/*********************************************************************/

DBUS<15:0>,                 ! External Data Bus
```

```
ABUS<23:1>;                   ! External Address Bus(changed)

format

/***********************************************************************/
/*                                                                     */
/*                        Register Subfields                           */
/*                                                                     */
/***********************************************************************/

PCADDR      = PC<23:0>,       ! Program Counter Address Field
SRTRACE     = SR<15>,         ! Trace Bit
SRMODE      = SR<13>,         ! Mode Selection Bit
SRCARRY     = SR<0>,          ! Carry Bit
SROVER      = SR<1>,          ! Overflow Bit
SRZERO      = SR<2>,          ! Zero Bit
SRNEG       = SR<3>,          ! Negative Bit
SREX        = SR<4>,          ! Extend Bit
SRMASK      = SR<10:8>,       ! Interrupt Mask
FCSPACE     = FC<1:0>,        ! Memory Access Address Space
FCMODE      = FC<2>,          ! User/Supervisor Mode Bit
PCLOW       = PC<15:0>,       ! PC Low Word
PCHI        = PC<31:16>,      ! PC High Word
D0LWORD     = D[0]<15:0>,     ! D[0] Low Word
D1LWORD     = D[1]<15:0>,     ! D[1] Low Word
D2LWORD     = D[2]<15:0>,     ! D[2] Low Word
D3LWORD     = D[3]<15:0>,     ! D[3] Low Word
D4LWORD     = D[4]<15:0>,     ! D[4] Low Word
D5LWORD     = D[5]<15:0>,     ! D[5] Low Word
D6LWORD     = D[6]<15:0>,     ! D[6] Low Word
D7LWORD     = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW   = HANADR<15:0>, ! HANADR Low Word
HANADRHI    = HANADR<31:16>,! HANADR High Word
TEMPADRLOW  = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI   = TEMPADR<31:16>;! TEMPADR High Word

memory

/***********************************************************************/
/*                                                                     */
/*                   16K 16-Bit Word Internal Memory                   */
/*                                                                     */
/***********************************************************************/

M[0:32767]<7:0>;

macro

/***********************************************************************/
/*                                                                     */
/*                      Logic Level Macros                             */
/*                                                                       */
/*************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*************************************************************************/
/*                                                                       */
/*  Power On and Initialization. This process was not modeled but is    */
/*  added to initialize signals and registers.                           */
/*                                                                       */
/*************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                    ! Turn Power On
        next;                           ! Execute Assignment
        READY = lo;                     ! System Not Ready
        RESET = lo;                     ! Assert Reset For
        delay(100);                     ! 100 Miliseconds(Active Low)
        RESET = hi;                     ! Deactivate Reset
        next;                           ! Execute Pending Assignments
        ASN = hi;                       ! Initialize Address Strobe
        LDSN = hi;                      ! Initialize Lower Data Strobe
        UDSN = hi;                      ! Initialize Upper Data Strobe
        DTACKN = hi;                    ! Initialize Data Transfer Acknowledge
        RW = hi;                        ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                 ! Place Data Bus In High Impedance State
        ABUS = 0xffffff;            ! Place Address Bus In High Impedance State
        M[0x100c] = 0xff;               ! Place Memory Locations Following The
        M[0x100d] = 0xff;               ! JMP Instruction In A High State
        HALT = hi;                      ! Initialize Halt Flip-Flop(Active
                                        ! Low)
        T = 0;                          ! Initialize Clock Cycle Counter
        READY = hi;                     ! System Ready
        /*************************************************************/
        /*                                                           */
        /*     Routine Initialization Per Hamby and Guillory         */
        /*                                                           */
        /*************************************************************/
        SRMODE = lo;                    ! Set Status Register To User Mode
        D[1] = 0x0;                     ! Place Hex 0 Into D[1]
        A[0] = 0x1004;                  ! Place Hex 1000 Into A[0]
        PC = 0x1004;                    ! Place Hex 1000 Into Program Counter
        M[0x2002] = 0x4e;               ! Place RTE Instruction In
        M[0x2003] = 0x73;               ! Location 2000 Hex
        M[0x10] = 0x0;                  ! Place Exception
        M[0x11] = 0x0;                  ! Vector In
        M[0x12] = 0x20;                 ! Location
        M[0x13] = 0x02;                 ! 10 Hex
        SA7 = 0x08e6;                   ! Initialize System Stack Pointer
                                        ! At Program Headed Down (Added)
        next                            ! Execute Assignments
        )

/************************************************************************/
/*                                                                      */
/*  Initial Fetcn Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/************************************************************************/

fetch_initial_instruction :=
      (

      /******************************************************************/

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 0
      RW = hi;                            ! Memory Read
      ADENABLE = lo;                      ! Disable Address Bus Buffer
      DBENABLE = lo;                      ! Disable Data Bus Buffer
      IABUS = PC;                         ! Place PC On Internal Address
                                          ! Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2 Of
      PHI2 = hi;                          ! Clock Cycle 0
      ADENABLE = hi;                      ! Enable Address Bus Buffer
      EXABUF = IABUS;                     ! Gate Internal Address Bus
                                          ! Into External Address Buffer
      FCMODE = SRMODE;                    ! User Mode
      FCSPACE = 2;                        ! Accessing Program
      next;                               ! Execute Impending Assignments
      ABUS = EXABUF;                      ! Address Placed On Bus(Added)
      next;                               ! Execute Pending Assignments

      /******************************************************************/
      T = 1;                              ! Clock Cycle 1
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1 Of
      PHI2 = lo;                          ! Clock Cycle 1
      ASN = lo;                           ! Assert Address Strobe
      LDSN = lo;                          ! Assert Lower Data Strobe
      UDSN = lo;                          ! Assert Upper Data Strobe
      DBENABLE = hi;                      ! Enable Data Bus
      next;                               ! Execute Pending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 1
```

```
next;                                    ! Execute Pending Assignments

/***************************************************************/
T = 2;                                   ! Clock Cycle 2
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 2
      next;                              ! Execute Assignments

      /*********************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*********************************************************/
      T = 2                              ! Return To  Phase 2
                                         ! Of Clock Cycle 2
      );
      next;                              ! Execute Impending Assignments

/***************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/***************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
PFR = EXDBUF;                            ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
```

```
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

   illegal :=                                   ! Illegal Instruction (4AFC)
          (

          /*******************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        ABUS = 0xffffff;                        ! Address Bus High Impedanced
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        DBUS = 0xffff;                          ! Data Bus High Impedanced
        IABUS<31:1> = PC<31:1>;                 ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        ABUS = IABUS<23:1>;                     ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /*******************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        DBENABLE = hi;                          ! Enable Data Bus
```

```
next;                                 ! Execute Pending Assignments

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 1
IDBUS = SR;                           ! Place Status Register On
                                      ! Internal Data Bus
PC = PC - 2;                         ! Decrement PC To Illegal Instruction
                                      ! Address (Changed)
next;                                 ! Execute Pending Assignments

/*****************************************************************/
T = 2;                                ! Clock Cycle 2
next;                                 ! Execute Assignment

PHI1 = hi;                            ! Phase 1
PHI2 = lo;                            ! Of Clock Cycle 2
SRTEMP = IDBUS;                       ! Place Status Register
                                      ! On Bus In Temporary Register
while DTACKN eql hi                   ! Wait For Memory To Place
      (                               ! Data On The Bus
      next;                           ! Execute Impending Assignments

      PHI1 = lo;                      ! Phase 2
      PHI2 = hi;                      ! Of Clock Cycle 2
      next;                           ! Execute Assignments

      /**********************************************************/
      T = 3;                          ! Clock Cycle 3
      next;                           ! Execute Assignment

      PHI1 = hi;                      ! Phase 1
      PHI2 = lo;                      ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];           ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];        ! On Data Bus And
      DTACKN = lo;                    ! Asserts DTACKN(Added)
      next;                           ! Execute Pending Assignments

      /**********************************************************/
      T = 2                           ! Return To  Phase 2
                                      ! Of Clock Cycle 2
      );
      next;                           ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                ! Clock Cycle 3
next;                                 ! Execute Assignment

PHI1 = lo;                            ! Phase 2
PHI2 = hi;                            ! Of Clock Cycle 3
EXDBUF = DBUS;                        ! Instruction On Data Bus
                                      ! Is Placed In External Data
                                      ! Bus Buffer
SRMODE = hi;                          ! Set Supervisor State
```

```
SRTRACE = lo;                                      ! Turn Off Trace
next;                                              ! Execute Pending Assignments

/******************************************************************/
T = 4;                                             ! Clock Cycle 4
next;                                              ! Execute Assignment

PHI1 = hi;                                         ! Phase 1
PHI2 = lo;                                         ! Of Clock Cycle 4
PFR = EXDBUF;                                      ! The Contents Of The External
                                                   ! Data Bus Buffer Are Placed
                                                   ! In Prefetch Register
SA7 = SA7 - 2;                                     ! Decrement System Stack Pointer
                                                   ! To Point To Location That Will
                                                   ! Receive PC's Low Word
next;                                              ! Execute Pending Assignments

PHI1 = lo;                                         ! Phase 2
PHI2 = hi;                                         ! Of Clock Cycle 4
ASN = hi;                                          ! Deactivate Address Strobe
LDSN = hi;                                         ! Deactivate Lower Data Strobe
UDSN = hi;                                         ! Deactivate Upper Data Strobe
DTACKN = hi;                                       ! Deactivate Data Transfer(Added)
                                                   ! Acknowledge
next;                                              ! Execute Pending Assignments

/******************************************************************/
T = 5;                                             ! Clock Cycle 5
next;

PHI1 = hi;                                         ! Phase 1 Of
PHI2 = lo;                                         ! Clock Cycle 5
ADENABLE = lo;                                     ! Disable Address Bus Buffer
ABUS = 0xffffff;                                   ! Address Bus High Impedanced
DBENABLE = lo;                                     ! Disable Data Bus Buffer
DBUS = 0xffff;                                     ! Data Bus High Impedanced
VECADR = 4;                                        ! Place Vector Number In Register
next;                                              ! Execute Pending Assignments

PHI1 = lo;                                         ! Phase 2 Of
PHI2 = hi;                                         ! Clock Cycle 5
next;                                              ! Execute Impending Assignments

/******************************************************************/
T = 6;                                             ! Clock Cycle 6
next;                                              ! Execute Assignment

PHI1 = hi;                                         ! Phase 1 Of
PHI2 = lo;                                         ! Clock Cycle 6
VECADR = VECADR *:arith 2;                         ! Multiply Vector Number
                                                   ! By 4 For Vector Address
next;                                              ! Execute Pending Assignments
```

```
PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 6
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                      ! Clock Cycle 7
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
next;

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 7
next;

/*****************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 8
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 8
next;

/*****************************************************************/
T = 9;                                      ! Clock Cycle 9
next;                                       ! Execute Assignment


PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 9
RW = hi;                                    ! Memory Read
ADENABLE = lo;                              ! Disable Address Bus Buffer
DBENABLE = lo;                              ! Disable Data Bus Buffer
DBUS = 0xffff;                              ! Data Bus High Impedanced
IABUS = SA7;                                 ! Place SA7 On Internal Address
                                            ! Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2 Of
PHI2 = hi;                                  ! Clock Cycle 9
ADENABLE = hi;                              ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;                       ! Gate Internal Address Bus
                                            ! Into External Address Buffer
FCMODE = SRMODE;                            ! Supervisor Mode
FCSPACE = 1;                                ! Accessing Data
IDBUS = PCLOW;                              ! Place Low Word from PC Onto
                                            ! Internal Data Bus
ABUS = IABUS<23:1>;                         ! Place Address On External Data Bus
```

```
next;                                   ! Execute Impending Assignments

/*************************************************************/
T = 10;                                  ! Clock Cycle 10
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 10
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Place Internal Data Bus
                                        ! Contents Into External Data Buffer
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 10
DBUS = EXDBUF;                          ! Contents Of External Data Buffer
                                        ! Placed On Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 11;                                  ! Clock Cycle 11
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 11
UDSN = lo;                              ! Activate Upper Data Strobe
LDSN = lo;                               ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      twait = twait + 1;
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 11
      next;                             ! Execute Assignments

      /*******************************************************/
      T = 12;                            ! Clock Cycle 12
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 12
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;             ! PC Low Word
      M[ABUS + 1] = DBUS<7:0>;           ! Stored
      DTACKN = lo                      ! Asserts DTACKN(Added)
      );
      next;                             ! Execute Pending Assignments
      /*********************************************************/
      T = 11                              ! Return To  Phase 2
                                          ! Of Clock Cycle 11
      );
      next;                               ! Execute Impending Assignments

/***************************************************************/
T = 12;                                   ! Clock Cycle 12
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 12
SA7 = SA7 - 4;                            ! Set System Stack Pointer
                                          ! To Point To Status Register
                                          ! Storage Location
next;                                     ! Execute Pending Assignments

/***************************************************************/
T = 13;                                   ! Clock Cycle 13
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 13
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 13
ASN = hi;                                 ! Deactivate Address Strobe
LDSN = hi;                                ! Deactivate Lower Data Strobe
UDSN = hi;                                ! Deactivate Upper Data Strobe
DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                          ! Acknowledge
next;                                     ! Execute Pending Assignments

/***************************************************************/
T = 14;                                   ! Clock Cycle 14
next;

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 14
RW = hi;                                  ! Memory Read
ADENABLE = lo;                            ! Disable Address Bus Buffer
ABUS = 0xffffff;                          ! Address Bus High Impedanced
DBENABLE = lo;                            ! Disable Data Bus Buffer
IABUS = SA7;                              ! Place SA7 On Internal Address
                                          ! Bus
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2 Of
PHI2 = hi;                                ! Clock Cycle 14
ADENABLE = hi;                            ! Enable Address Bus Buffer
DBUS = 0xffff;                            ! Data Bus High Impedanced
```

```
EXABUF = IABUS<23:1>;                   ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
IDBUS = SRTEMP;                         ! Place Holder Of Status Register
                                        ! Onto Internal Data Bus
ABUS = IABUS<23:1>;                     ! Place Address On Address Bus
next;                                   ! Execute Impending Assignments

/*****************************************************************/
T = 15;                                  ! Clock Cycle 15
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 15
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Internal Data Bus Moved
                                        ! To External Data Buffer
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 15
DBUS = EXDBUF;                          ! Contents Of External Data
                                        ! Buffer Placed On Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 16;                                  ! Clock Cycle 16
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 16
UDSN = lo;                              ! Activate Upper Data Strobe
LDSN = lo;                               ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      twait = twait + 1;
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 16
      next;                             ! Execute Assignments

      /***********************************************************/
      T = 17;                            ! Clock Cycle 17
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 17
```

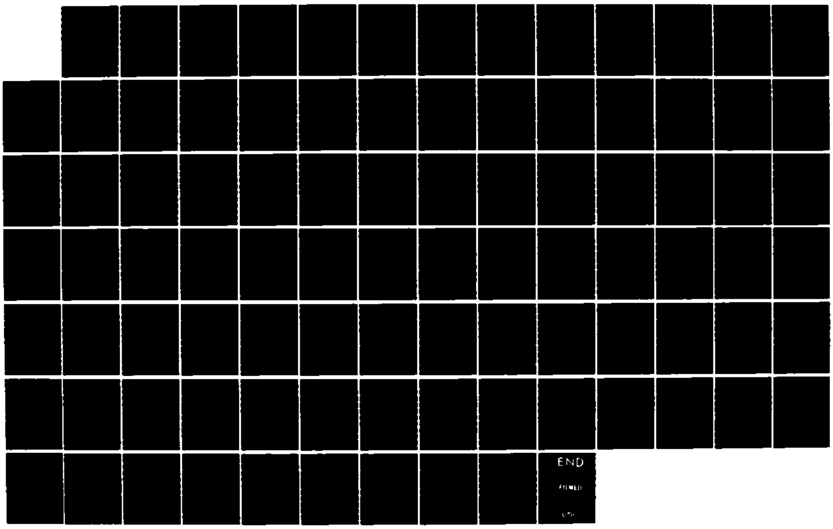
```
        if twait eql 2
        (
        M[ABUS] = DBUS<15:8>;              ! PC Low Word
        M[ABUS + 1] = DBUS<7:0>;            ! Stored
        DTACKN = lo                       ! Asserts DTACKN(Added)
        );
        next;                              ! Execute Pending Assignments

        /*********************************************************/
        T = 16                              ! Return To  Phase 2
                                           ! Of Clock Cycle 16
        );
        next;                              ! Execute Impending Assignments

/*****************************************************************/
T = 17;                                     ! Clock Cycle 17
next;                                      ! Execute Assignment

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 17
SA7 = SA7 + 2;                             ! Set System Stack Pointer
                                           ! To Point To High PC
                                           ! Storage Location
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 18;                                     ! Clock Cycle 18
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 18
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 18
ASN = hi;                                  ! Deactivate Address Strobe
LDSN = hi;                                 ! Deactivate Lower Data Strobe
UDSN = hi;                                 ! Deactivate Upper Data Strobe
DTACKN = hi;                               ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
next;                                      ! Execute Pending Assignments

/*****************************************************************/
T = 19;                                    ! Clock Cycle 19
next;                                      ! Execute Pending Assignments

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 19
RW = hi;                                   ! Memory Read
ADENABLE = lo;                             ! Disable Address Bus Buffer
ABUS = 0xffffff;                           ! Address Bus High Impedanced
DBENABLE = lo;                             ! Disable Data Bus Buffer
IABUS = SA7;                               ! Place SA7 On Internal Address
```

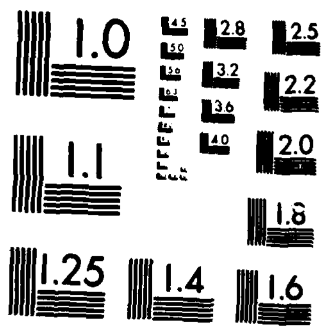
```
                                                   ! Bus
next;                                              ! Execute Pending Assignments

PHI1 = lo;                                         ! Phase 2 Of
PHI2 = hi;                                         ! Clock Cycle 19
ADENABLE = hi;                                     ! Enable Address Bus Buffer
DBUS = 0xffff;                                     ! Data Bus High Impedanced
EXABUF = IABUS<23:1>;                              ! Gate Internal Address Bus
                                                   ! Into External Address Buffer
FCMODE = SRMODE;                                   ! Supervisor Mode
FCSPACE = 1;                                       ! Accessing Data
IDBUS = PCHI;                                      ! Place High Word Of PC
                                                   ! Onto Internal Data Bus
ABUS = IABUS<23:1>;                                ! Place Address On External Bus
next;                                              ! Execute Impending Assignments

/*****************************************************************/
T = 20;                                             ! Clock Cycle 20
next;                                              ! Execute Assignment

PHI1 = hi;                                         ! Phase 1 Of
PHI2 = lo;                                         ! Clock Cycle 20
ASN = lo;                                          ! Assert Address Strobe
RW = lo;                                           ! Activate Write
EXDBUF = IDBUS;                                    ! Internal Data Bus Moved
                                                   ! To External Data Buffer
next;                                              ! Execute Pending Assignments

PHI1 = lo;                                         ! Phase 2
PHI2 = hi;                                         ! Of Clock Cycle 20
DBUS = EXDBUF;                                     ! Contents Of External Data
                                                   ! Buffer Placed On Data Bus
DBENABLE = hi;                                     ! Enable Data Bus
next;                                              ! Execute Pending Assignments

/*****************************************************************/
T = 21;                                             ! Clock Cycle 21
next;                                              ! Execute Assignment

PHI1 = hi;                                         ! Phase 1
PHI2 = lo;                                         ! Of Clock Cycle 21
UDSN = lo;                                         ! Activate Upper Data Strobe
LDSN = lo;                                          ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                                ! Wait For Memory To Place
      (                                            ! Data On The Bus
      twait = twait + 1;
      next;                                        ! Execute Impending Assignments

      PHI1 = lo;                                   ! Phase 2
      PHI2 = hi;                                   ! Of Clock Cycle 21
      next;                                        ! Execute Assignments
```

```
          /**************************************************************/
          T = 22;                                ! Clock Cycle 22
          next;                                 ! Execute Assignment

          PHI1 = hi;                            ! Phase 1
          PHI2 = lo;                            ! Of Clock Cycle 22
          if twait eql 2
          (
          M[ABUS] = DBUS<15:8>;                 ! PC Low Word
          M[ABUS + 1] = DBUS<7:0>;               ! Stored
          DTACKN = lo                          ! Asserts DTACKN(Added)
          );
          next;                                 ! Execute Pending Assignments

          /**************************************************************/
          T = 21                                 ! Return To  Phase 2
                                                ! Of Clock Cycle 21
          );
          next;                                 ! Execute Impending Assignments

    /********************************************************************/
    T = 22;                                      ! Clock Cycle 22
    next;                                       ! Execute Assignment

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 22
    next;                                       ! Execute Pending Assignments

    /********************************************************************/
    T = 23;                                      ! Clock Cycle 23
    next;                                       ! Execute Assignment

    PHI1 = hi;                                  ! Phase 1
    PHI2 = lo;                                  ! Of Clock Cycle 23
    next;                                       ! Execute Pending Assignments

    PHI1 = lo;                                  ! Phase 2
    PHI2 = hi;                                  ! Of Clock Cycle 23
    ASN = hi;                                   ! Deactivate Address Strobe
    LDSN = hi;                                  ! Deactivate Lower Data Strobe
    UDSN = hi;                                  ! Deactivate Upper Data Strobe
    DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
    next;                                       ! Execute Pending Assignments

    /********************************************************************/

    T = 24;                                      ! Clock Cycle 24
    next;                                       ! Execute Assignment

    PHI1 = hi;                                  ! Phase 1
    PHI2 = lo;                                  ! Of Clock Cycle 24
```

AD-A164 257 THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. C A BAXLEY 5/5

UNCLASSIFIED DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
RW = hi;                                   ! Memory Read
ADENABLE = lo;                             ! Disable Address Bus Buffer
ABUS = 0xffffff;                           ! Address Bus High Impedanced
DBENABLE = lo;                             ! Disable Data Bus Buffer
IABUS = VECADR;                            ! Place VECADR On Internal Address
                                           ! Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2 Of
PHI2 = hi;                                 ! Clock Cycle 24
ADENABLE = hi;                             ! Enable Address Bus Buffer
DBUS = 0xffff;                             ! Data Bus High Impedanced
EXABUF = IABUS<23:1>;                      ! Gate Internal Address Bus
                                           ! Into External Address Buffer
FCMODE = SRMODE;                           ! Supervisor Mode
FCSPACE = 1;                               ! Accessing Data
ABUS = IABUS<23:1>;                        ! Place Address On External Bus
next;                                      ! Execute Impending Assignments

/*************************************************************/
T = 25;                                     ! Clock Cycle 25
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1 Of
PHI2 = lo;                                 ! Clock Cycle 25
ASN = lo;                                  ! Assert Address Strobe
LDSN = lo;                                 ! Assert Lower Data Strobe
UDSN = lo;                                 ! Assert Upper Data Strobe
DBENABLE = hi;                             ! Enable Data Bus
next;                                      ! Execute Pending Assignments

PHI1 = lo;                                 ! Phase 2
PHI2 = hi;                                 ! Of Clock Cycle 25
next;                                      ! Execute Pending Assignments

/*************************************************************/
T = 26;                                     ! Clock Cycle 26
next;                                      ! Execute Assignment

PHI1 = hi;                                 ! Phase 1
PHI2 = lo;                                 ! Of Clock Cycle 26
while DTACKN eql hi                        ! Wait For Memory To Place
      (                                    ! Data On The Bus
      next;                                ! Execute Impending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 26
      next;                                ! Execute Assignments

      /******************************************************/
      T = 27;                               ! Clock Cycle 27
      next;                                ! Execute Assignment
```

```
          PHI1 = hi;                          ! Phase 1
          PHI2 = lo;                          ! Of Clock Cycle 27
          DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
          DTACKN = lo;                        ! Asserts DTACKN(Added)
          next;                               ! Execute Pending Assignments

          /**********************************************************/
          T = 26                               ! Return To  Phase 2
                                              ! Of Clock Cycle 26
          );
          next;                               ! Execute Impending Assignments

     /****************************************************************/
     T = 27;                                   ! Clock Cycle 27
     next;                                    ! Execute Assignment

     PHI1 = lo;                               ! Phase 2
     PHI2 = hi;                               ! Of Clock Cycle 27
     EXDBUF = DBUS;                           ! Instruction On Data Bus
                                              ! Is Placed In External Data
                                              ! Bus Buffer
     next;                                    ! Execute Pending Assignments

     /****************************************************************/
     T = 28;                                   ! Clock Cycle 28
     next;                                    ! Execute Assignment

     PHI1 = hi;                               ! Phase 1
     PHI2 = lo;                               ! Of Clock Cycle 28
     HANADRLOW = EXDBUF;                      ! The Contents Of The External
                                              ! Data Bus Buffer Are Placed
                                              ! In Handler Routine Low Address
     next;                                    ! Execute Pending Assignments

     PHI1 = lo;                               ! Phase 2
     PHI2 = hi;                               ! Of Clock Cycle 28
     ASN = hi;                                ! Deactivate Address Strobe
     LDSN = hi;                               ! Deactivate Lower Data Strobe
     UDSN = hi;                               ! Deactivate Upper Data Strobe
     DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                              ! Acknowledge
     VECADR = VECADR + 2;                     ! Increment Vector Address Register
                                              ! To Pick Handler Address Low Word
     next;                                    ! Execute Pending Assignments

     /****************************************************************/

     T = 29;                                   ! Clock Cycle 29
     next;                                    ! Execute Assignment

     PHI1 = hi;                               ! Phase 1
     PHI2 = lo;                               ! Of Clock Cycle 24
```

```
RW = hi;                            ! Memory Read
ADENABLE = lo;                      ! Disable Address Bus Buffer
ABUS = 0xffffff;                    ! Address Bus High Impedanced
DBENABLE = lo;                      ! Disable Data Bus Buffer
DBUS = 0xffff;                      ! Data Bus High Impedanced
IABUS = VECADR;                     ! Place VECADR On Internal Address
                                    ! Bus
IDBUS = HANADRLOW;                  ! Move Handler High Address To
                                    ! Data Bus
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2 Of
PHI2 = hi;                          ! Clock Cycle 29
ADENABLE = hi;                      ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;               ! Gate Internal Address Bus
                                    ! Into External Address Buffer
FCMODE = SRMODE;                    ! Supervisor Mode
FCSPACE = 1;                        ! Accessing Data
HANADRHI = IDBUS;                   ! Move Handler High Address
                                    ! To Upper Word Of Register
ABUS = IABUS<23:1>;                 ! Place Address On External Bus
next;                               ! Execute Impending Assignments

/****************************************************************/
T = 30;                              ! Clock Cycle 30
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1 Of
PHI2 = lo;                          ! Clock Cycle 30
ASN = lo;                           ! Assert Address Strobe
LDSN = lo;                          ! Assert Lower Data Strobe
UDSN = lo;                          ! Assert Upper Data Strobe
DBENABLE = hi;                      ! Enable Data Bus
next;                               ! Execute Pending Assignments

PHI1 = lo;                          ! Phase 2
PHI2 = hi;                          ! Of Clock Cycle 30
next;                               ! Execute Pending Assignments

/****************************************************************/
T = 31;                              ! Clock Cycle 31
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1
PHI2 = lo;                          ! Of Clock Cycle 31
while DTACKN eql hi                 ! Wait For Memory To Place
      (                             ! Data On The Bus
      next;                         ! Execute Impending Assignments

      PHI1 = lo;                    ! Phase 2
      PHI2 = hi;                    ! Of Clock Cycle 31
      next;                         ! Execute Assignments
```

```
          /*************************************************************/
          T = 32;                                 ! Clock Cycle 32
          next;                                   ! Execute Assignment

          PHI1 = hi;                              ! Phase 1
          PHI2 = lo;                              ! Of Clock Cycle 32
          DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
          DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
          DTACKN = lo;                            ! Asserts DTACKN(Added)
          next;                                   ! Execute Pending Assignments

          /*************************************************************/
          T = 31                                  ! Return To  Phase 2
                                                  ! Of Clock Cycle 31
          );
          next;                                   ! Execute Impending Assignments

      /*****************************************************************/
      T = 32;                                     ! Clock Cycle 32
      next;                                       ! Execute Assignment

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 32
      EXDBUF = DBUS;                              ! Instruction On Data Bus
                                                  ! Is Placed In External Data
                                                  ! Bus Buffer
      next;                                       ! Execute Pending Assignments

      /*****************************************************************/
      T = 33;                                     ! Clock Cycle 33
      next;                                       ! Execute Assignment

      PHI1 = hi;                                  ! Phase 1
      PHI2 = lo;                                  ! Of Clock Cycle 33
      HANADRLOW = EXDBUF;                         ! The Contents Of The External
                                                  ! Data Bus Buffer Are Placed
                                                  ! In Handler Routine Low Address
      next;                                       ! Execute Pending Assignments

      PHI1 = lo;                                  ! Phase 2
      PHI2 = hi;                                  ! Of Clock Cycle 33
      ASN = hi;                                   ! Deactivate Address Strobe
      LDSN = hi;                                  ! Deactivate Lower Data Strobe
      UDSN = hi;                                  ! Deactivate Upper Data Strobe
      DTACKN = hi;                                ! Deactivate Data Transfer(Added)
                                                  ! Acknowledge
      next;                                       ! Execute Pending Assignments

      /*****************************************************************/

      T = 34;                                     ! Clock Cycle 34
      next;                                       ! Execute Assignment
```

```
PHI1 = hi;                                   ! Phase 1 Of
PHI2 = lo;                                   ! Clock Cycle 34
RW = hi;                                     ! Memory Read
ADENABLE = lo;                               ! Disable Address Bus Buffer
ABUS = 0xffffff;                             ! Address Bus High Impedanced
DBENABLE = lo;                               ! Disable Data Bus Buffer
DBUS = 0xffff;                               ! Data Bus High Impedanced
IABUS = HANADR;                              ! Place HANADR On Internal Address
                                             ! Bus
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2 Of
PHI2 = hi;                                   ! Clock Cycle 34
ADENABLE = hi;                               ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>                         ! Gate Internal Address Bus
                                             ! Into External Address Buffer
FCMODE = SRMODE;                             ! User Mode
PC = IABUS;                                  ! Place HANADR In PC
FCSPACE = 2;                                 ! Accessing Program
ABUS = IABUS<23:1>;                          ! Place Address On External Bus
next;                                        ! Execute Impending Assignments

/*************************************************************/
T = 35;                                       ! Clock Cycle 35
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1 Of
PHI2 = lo;                                   ! Clock Cycle 35
ASN = lo;                                    ! Assert Address Strobe
LDSN = lo;                                   ! Assert Lower Data Strobe
UDSN = lo;                                   ! Assert Upper Data Strobe
DBENABLE = hi;                               ! Enable Data Bus
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 35
next;                                        ! Execute Pending Assignments

/*************************************************************/
T = 36;                                       ! Clock Cycle 36
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1
PHI2 = lo;                                   ! Of Clock Cycle 36
while DTACKN eql hi                          ! Wait For Memory To Place
      (                                      ! Data On The Bus
      next;                                  ! Execute Impending Assignments

      PHI1 = lo;                             ! Phase 2
      PHI2 = hi;                             ! Of Clock Cycle 36
      next;                                  ! Execute Assignments

      /*******************************************************/
```

```
            T = 37;                             ! Clock Cycle 37
            next;                              ! Execute Assignment

            PHI1 = hi;                         ! Phase 1
            PHI2 = lo;                         ! Of Clock Cycle 37
            DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
            DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
            DTACKN = lo;                       ! Asserts DTACKN(Added)
            next;                              ! Execute Pending Assignments

            /*****************************************************************/
            T = 36                              ! Return To  Phase 2
                                               ! Of Clock Cycle 36
            );
            next;                              ! Execute Impending Assignments

      /***********************************************************************/
      T = 37;                                   ! Clock Cycle 37
      next;                                    ! Execute Assignment

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 37
      EXDBUF = DBUS;                           ! Instruction On Data Bus
                                               ! Is Placed In External Data
                                               ! Bus Buffer
      next;                                    ! Execute Pending Assignments

      /***********************************************************************/
      T = 38;                                   ! Clock Cycle 38
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 38
      PFR = EXDBUF;                            ! The Contents Of The External
                                               ! Data Bus Buffer Are Placed
                                               ! In Prefetch Register
      next;                                    ! Execute Pending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 38
      ASN = hi;                                ! Deactivate Address Strobe
      LDSN = hi;                               ! Deactivate Lower Data Strobe
      UDSN = hi;                               ! Deactivate Upper Data Strobe
      IR = PFR;                                ! Contents Of Prefetch Register
                                               ! Are Placed Into Instruction
                                               ! Register
      DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                               ! Acknowledge
      PC = PC + 2;                             ! Increment Program Counter
      next;                                    ! Execute Pending Assignments

      /***********************************************************************/
```

```
        T = 39;                                 ! Clock Cycle 39
        next;

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 39
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        DBUS = 0xffff;                          ! Data Bus High Impedanced
        ASN = hi;                               ! Disable Address,
        LDSN = hi;                              ! Lower Data, and
        UDSN = hi;                              ! Upper Data Strobes
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 39
        ABUS = 0xffffff;                        ! ABUS High Impedanced
        next;                                   ! Execute Impending Assignments

        /*************************************************************/
        T = 40;                                  ! Clock Cycle 40
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 40
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 40
        next;

        T = 0                                   ! Reset Clock Cycle Counter
        )

    rte :=
       (
       SA7 = SA7 - 2;
       next;
       SR<15:8> = M[SA7];
       SR<7:0> = M[SA7 + 1];
       next;
       SA7 = SA7 + 2;
       next;
       PC<31:24> = M[SA7];
       PC<23:16> = M[SA7 + 1];
       next;
       SA7 = SA7 + 2;
       next;
       PC<15:8> = M[SA7];
       PC<7:0> = M[SA7 + 1];
       next;
       IR<15:8> = M[PC];
       IR<7:0> = M[PC + 1];
       next;
```

```
     PC = PC + 2;
     T = 20;
     next;
     T = 0
     )

move :=                                      ! MOVE.W D1,D2
       (

      /*************************************************************/

      PHI1 = hi;                             ! Phase 1 Of
      PHI2 = lo;                             ! Clock Cycle 0
      DBUS = 0xffff;                         ! Place Data Bus In High Impedance
      RW = hi;                               ! Memory Read
      ADENABLE = lo;                         ! Disable Address Bus Buffer
      DBENABLE = lo;                         ! Disable Data Bus Buffer
      IABUS = PC;                            ! Place PC On Internal Address
                                             ! Bus
      IDBUS = D1LWORD;                       ! Place Low Word From D[1] Onto
                                             ! Internal Data Bus
      next;                                  ! Execute Pending Assignments

      PHI1 = lo;                             ! Phase 2 Of
      PHI2 = hi;                             ! Clock Cycle 0
      ADENABLE = hi;                         ! Enable Address Bus Buffer
      EXABUF = IABUS;                        ! Gate Internal Address Bus
                                             ! Into External Address Buffer
      FCMODE = SRMODE;                       ! User Mode
      FCSPACE = 2;                           ! Accessing Program
      SRCARRY = lo;                          ! Clear Status Register Carry Bit
      SROVER = lo;                           ! Clear Status Register Overflow Bit
      SRZERO = lo;                           ! Clear Status Register Zero Bit
      SRNEG  = lo;                           ! Clear Status Register Negative Bit
      D2LWORD = IDBUS;                       ! Place Data From Internal Data Bus
                                             ! Into Low Word Of D[2]
      next;                                  ! Execute Impending Assignments
      ABUS = EXABUF;                         ! Address Placed On Bus(Added)
      next;                                  ! Execute Pending Assignments

      /*************************************************************/
      T = 1;                                 ! Clock Cycle 1
      next;                                  ! Execute Assignment

      PHI1 = hi;                             ! Phase 1 Of
      PHI2 = lo;                             ! Clock Cycle 1
      ASN = lo;                              ! Assert Address Strobe
      LDSN = lo;                             ! Assert Lower Data Strobe
      UDSN = lo;                             ! Assert Upper Data Strobe
      DBENABLE = hi;                         ! Enable Data Bus
      if D2LWORD eql 0                       ! Set Status Register Zero Bit
         SRZERO = hi;                        ! If Moved Data Is Zero
      next;                                  ! Execute Pending Assignments
```

```
PHI1 = lo;                          ! Phase 2
PHI2 = hi;                          ! Of Clock Cycle 1
if D[2]<15>                         ! Set Status Register Negative
   SRNEG = hi;                      ! Bit If Moved Data Is Negative
next;                               ! Execute Pending Assignments

/*******************************************************************/
T = 2;                              ! Clock Cycle 2
next;                               ! Execute Assignment

PHI1 = hi;                          ! Phase 1
PHI2 = lo;                          ! Of Clock Cycle 2
while DTACKN eql hi                 ! Wait For Memory To Place
      (                             ! Data On The Bus
      next;                         ! Execute Impending Assignments

      PHI1 = lo;                    ! Phase 2
      PHI2 = hi;                    ! Of Clock Cycle 2
      next;                         ! Execute Assignments

      /*************************************************************/
      T = 3;                        ! Clock Cycle 3
      next;                         ! Execute Assignment

      PHI1 = hi;                    ! Phase 1
      PHI2 = lo;                    ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];         ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];      ! On Data Bus And
      DTACKN = lo;                  ! Asserts DTACKN(Added)
      next;                         ! Execute Pending Assignments

      /*************************************************************/
      T = 2                         ! Return To  Phase 2
                                    ! Of Clock Cycle 2
      );
      next;                         ! Execute Impending Assignments

/*******************************************************************/
T = 3;                              ! Clock Cycle 3
next;                               ! Execute Assignment

PHI1 = lo;                          ! Phase 2
PHI2 = hi;                          ! Of Clock Cycle 3
EXDBUF = DBUS;                      ! Instruction On Data Bus
                                    ! Is Placed In External Data
                                    ! Bus Buffer
next;                               ! Execute Pending Assignments

/*******************************************************************/
T = 4;                              ! Clock Cycle 4
next;                               ! Execute Assignment
```

```
        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 4
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 4
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        IR = PFR;                               ! Contents Of Prefetch Register
                                                ! Are Placed Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
        PC = PC + 2;                            ! Increment Program Counter
        next;                                   ! Execute Impending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )


    jmp :=                                      ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High Impedance
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 1
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
IABUS = A[0];                           ! Move Jump Address From A[0]
                                        ! To Internal Address Buffer
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 1
PC = IABUS;                             ! Place Jump Address Into Program
                                        ! Counter
next;

/**************************************************************/
T = 2;                                  ! Clock Cycle 2
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 2
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 2
      next;                             ! Execute Assignments

      /********************************************************/
      T = 3;                            ! Clock Cycle 3
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments

      /********************************************************/
      T = 2                             ! Return To Phase 2
                                        ! Of Clock Cycle 2
      );
      next;                             ! Execute Impending Assignments

/**************************************************************/
T = 3;                                  ! Clock Cycle 3
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
```

```
PHI2 = hi;                             ! Of Clock Cycle 3
EXDBUF = DBUS;                         ! Instruction On Data Bus
                                       ! Is Placed In External Data
                                       ! Bus Buffer
next;                                  ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                 ! Clock Cycle 4
next;                                  ! Execute Assignment

PHI1 = hi;                             ! Phase 1
PHI2 = lo;                             ! Of Clock Cycle 4
next;
PFR = EXDBUF;                          ! The Contents Of The External
                                       ! Data Bus Buffer Are Placed
                                       ! In Prefetch Register
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2
PHI2 = hi;                             ! Of Clock Cycle 4
ASN = hi;                              ! Deactivate Address Strobe
LDSN = hi;                             ! Deactivate Lower Data Strobe
UDSN = hi;                             ! Deactivate Upper Data Strobe
DTACKN = hi;                           ! Deactivate Data Transfer
                                       ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                 ! Clock Cycle 5
next;                                  ! Execute Previous Assignment

PHI1 = hi;                             ! Phase 1 Of
PHI2 = lo;                             ! Clock Cycle 5
RW = hi;                               ! Memory Read
ADENABLE = lo;                         ! Disable Address Bus Buffer
DBENABLE = lo;                         ! Disable Data Bus Buffer
IABUS = PC;                            ! Place PC On Internal Address
                                       ! Bus
next;                                  ! Execute Pending Assignments

PHI1 = lo;                             ! Phase 2 Of
PHI2 = hi;                             ! Clock Cycle 5
ADENABLE = hi;                         ! Enable Address Bus Buffer
FCMODE = SRMODE;                       ! User Mode
FCSPACE = 2;                           ! Accessing Program
EXABUF = IABUS;                        ! Gate Internal Address Bus
next;                                  ! Into External Address Buffer
ABUS = EXABUF;                         ! Address Placed On Bus(Added)
next;                                  ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                 ! Clock Cycle 6
next;                                  ! Execute Assignment
```

```
PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 6
ASN = lo;                                ! Assert Address Strobe
LDSN = lo;                               ! Assert Lower Data Strobe
UDSN = lo;                               ! Assert Upper Data Strobe
DBENABLE = hi;                           ! Enable Data Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 6
next;                                    ! Execute Pending Assignments

/***************************************************************/
T = 7;                                   ! Clock Cycle 7
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 7
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 7
      next;                              ! Execute Assignments

      /*********************************************************/
      T = 8;                             ! Clock Cycle 8
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /*********************************************************/
      T = 7                             ! Return To Phase 2
                                        ! Of Clock Cycle 7
      );
      next;                             ! Execute Impending Assignments

/***************************************************************/
T = 8;                                   ! Clock Cycle 8
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 8
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments
```

```
        /**************************************************************/
        T = 9;                                  ! Clock Cycle 9
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1
        PHI2 = lo;                              ! Of Clock Cycle 9
        PFR = EXDBUF;                           ! The Contents Of The External
                                                ! Data Bus Buffer Are Placed
                                                ! In Prefetch Register
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 9
        ASN = hi;                               ! Deactivate Address Strobe
        LDSN = hi;                              ! Deactivate Lower Data Strobe
        UDSN = hi;                              ! Deactivate Upper Data Strobe
        PC = PC + 2;                            ! Increment Program Counter
        IR = PFR;                               ! Place Contents Of Prefetch
                                                ! Register Into Instruction
                                                ! Register
        DTACKN = hi;                            ! Deactivate Data Transfer
                                                ! Acknowledge(Added)
        next;                                   ! Execute Pending Assignments
        T = 0                                   ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                        (
                        case IR
                             032001: move    ! MOVE.W D1,D2
                             047320: jmp     ! JMP (A0) If IR = Octal Value
                             0x4e73: rte     ! RTE (Return From Exception)
                             0x4afc: illegal! Illegal Instruction
                        esac
                        )

    main :=
         (
         power_on_initialize;
         fetch_initial_instruction;
         while READY eql hi
               (
               decode_execute_prefetch
               )
         )
```

```
/**********************************************************************/
/*                                                                    */
/*     MOTOROLA MC68000 MODEL OF THE ILLEGAL ADDRESS EXCEPTION        */
/*                                                                    */
/**********************************************************************/

/**********************************************************************/
/*                                                                    */
/*                    Structure Declarations                          */
/*                                                                    */
/**********************************************************************/

state

/**********************************************************************/
/*                                                                    */
/*                 M68000 Programming Registers                       */
/*                                                                    */
/**********************************************************************/

D[0:7]<31:0>,                ! 8 Data Registers
A[0:6]<31:0>,                ! 7 Address Registers
UA7<31:0>,                   ! User Stack Pointer
SA7<31:0>,                   ! System Stack Pointer
PC<31:0>,                    ! Program Counter
SR<15:0>,                    ! Status Register

/**********************************************************************/
/*                                                                    */
/*                 Temporary Internal Registers                       */
/*                                                                    */
/**********************************************************************/

PFR<15:0>,                   ! Prefetch Register
IR<15:0>,                    ! Instruction Register
FC<2:0>,                     ! Function Code Register
EXDBUF<15:0>,                ! External Data Bus Buffer Register
EXABUF<23:1>,                ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,               ! ALU Buffer 1
ALUBUF2<31:0>,               ! ALU Buffer 2
DTEMP<15:0>,                 ! Temporary Data Storage
DISREG<31:0>,                ! Temporary Displacement Storage
SRTEMP<15:0>,                ! Temporary Status Register Storage
                             ! (Exception Processing)
IRTEMP<15:0>,                ! Temporary Instruction Register Storage
                             ! (Exception Processing)
TEMPADR<31:0>,               ! Temporary Cycle Address Storage
                             ! (Exception Processing)
ACTYPE<15:0>,                ! Temporary Access Type Storage
                             ! (Exception Processing)
VECADR<23:0>,                ! Temporary Vector Address Storage
                             ! (Exception Processing)
```

```
HANADR<31:0>,                  ! Temporary Address Storage For
                               ! Exception Handler Routine
T<7:0>,                        ! Clock Cycle Counter
RESET,                      ! Reset Flip-Flop
HALT,                       ! Halt Flip-Flop
RW,                         ! Read/Write Flip-Flop
ADENABLE,                   ! Address Bus Buffer Enable
DBENABLE,                   ! Data Bus Buffer Enable
ASN,                        ! Address Strobe Flip-Flop
LDSN,                       ! Lower Data Strobe Flip-Flop
UDSN,                       ! Upper Data Strobe Flip-Flop
DTACKN,                     ! Data Transfer Acknowledge Flip-Flop
COUT,                       ! Carry Flip-Flop
EXCEPT,                     ! Exception Processing Flip-Flop
READY,                      ! Ready Flip-Flop

/***********************************************************************/
/*                                                                     */
/*      Model transformation modifications:                            */
/*                                                                     */
/*           1) CDL decoder structure nonexistent in ISP' and un-      */
/*      necessary for model. Eliminated.                               */
/*           2) Multi-phase clock structure nonexistent in ISP'.       */
/*      Operations on registers will provide its equivalent.           */
/*           3) Switch structure nonexistent in ISP'. Operation on a   */
/*      register will provide its equivalent.                          */
/*           4) The declared bus structures are modeled with registers */
/*      without loss of model accurracy. This done to maintain model   */
/*      equivalency and simplicity.                                    */
/*           5) The memory word length was reduced from 16 to 8 bit    */
/*      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*      their PC incrementation, and to enable the use of existing     */
/*      MC68000 assembler and linker/loader models. The memory was     */
/*      also reduced from 8 Mwords to 32 Kbytes.                       */
/*                                                                     */
/***********************************************************************/

IABUS<31:0>,                   ! Internal Address Bus
IDBUS<31:0>,                   ! Internal Data Bus
twait<7:0>,                    ! Wait Cycle Counter
SWITCH,                     ! Power Switch
PHI1,                       ! Phase 1 Of Two-Phase Clock
PHI2;                       ! Phase 2 Of Two-Phase Clock

port

/***********************************************************************/
/*                                                                     */
/*                External Address and Data Bus                        */
/*                                                                     */
/***********************************************************************/

DBUS<15:0>,                    ! External Data Bus
```

```
ABUS<23:1>;                     ! External Address Bus(changed)

format

/*************************************************************************/
/*                                                                       */
/*                         Register Subfields                            */
/*                                                                       */
/*************************************************************************/

PCADDR       = PC<23:0>,       ! Program Counter Address Field
SRTRACE      = SR<15>,         ! Trace Bit
SRMODE       = SR<13>,         ! Mode Selection Bit
SRCARRY      = SR<0>,          ! Carry Bit
SROVER       = SR<1>,          ! Overflow Bit
SRZERO       = SR<2>,          ! Zero Bit
SRNEG        = SR<3>,          ! Negative Bit
SREX         = SR<4>,          ! Extend Bit
SRMASK       = SR<10:8>,       ! Interrupt Mask
FCSPACE      = FC<1:0>,        ! Memory Access Address Space
FCMODE       = FC<2>,          ! User/Supervisor Mode Bit
PCLOW        = PC<15:0>,       ! PC Low Word
PCHI         = PC<31:16>,      ! PC High Word
D0LWORD      = D[0]<15:0>,     ! D[0] Low Word
D1LWORD      = D[1]<15:0>,     ! D[1] Low Word
D2LWORD      = D[2]<15:0>,     ! D[2] Low Word
D3LWORD      = D[3]<15:0>,     ! D[3] Low Word
D4LWORD      = D[4]<15:0>,     ! D[4] Low Word
D5LWORD      = D[5]<15:0>,     ! D[5] Low Word
D6LWORD      = D[6]<15:0>,     ! D[6] Low Word
D7LWORD      = D[7]<15:0>,     ! D[7] Low Word
DISREGHWORD  = DISREG<31:16>,! DISREG High Word
DISREGLWORD  = DISREG<15:0>, ! DISREG Low Word
HANADRLOW    = HANADR<15:0>, ! HANADR Low Word
HANADRHI     = HANADR<31:16>,! HANADR High Word
TEMPADRLOW   = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI    = TEMPADR<31:16>;! TEMPADR High Word

memory

/*************************************************************************/
/*                                                                       */
/*                   16K 16-Bit Word Internal Memory                     */
/*                                                                       */
/*************************************************************************/

M[0:32767]<7:0>;

macro

/*************************************************************************/
/*                                                                       */
/*                       Logic Level Macros                              */
/*                                                                           */
/*****************************************************************************/

lo    = 0 &,
hi    = 1 &,
off   = 0 &,
on    = 1 &,
clear = 0 &;

/*****************************************************************************/
/*                                                                           */
/*  Power On and Initialization. This process was not modeled but is        */
/*  added to initialize signals and registers.                               */
/*                                                                           */
/*****************************************************************************/

power_on_initialize :=
        (
        SWITCH = on;                        ! Turn Power On
        next;                               ! Execute Assignment
        READY = lo;                         ! System Not Ready
        RESET = lo;                         ! Assert Reset For
        delay(100);                         ! 100 Miliseconds(Active Low)
        RESET = hi;                         ! Deactivate Reset
        next;                               ! Execute Pending Assignments
        ASN = hi;                           ! Initialize Address Strobe
        LDSN = hi;                          ! Initialize Lower Data Strobe
        UDSN = hi;                          ! Initialize Upper Data Strobe
        DTACKN = hi;                        ! Initialize Data Transfer Acknowledge
        RW = hi;                            ! Initialize Read/Write(Read On High)
        DBUS = 0xffff;                     ! Place Data Bus In High Impedance State
        M[0x100a] = 0xff;                   ! Place Memory Locations Following The
        M[0x100b] = 0xff;                   ! JMP Instruction In A High State
        HALT = hi;                          ! Initialize Halt Flip-Flop(Active
                                            ! Low)
        T = 0;                              ! Initialize Clock Cycle Counter
        READY = hi;                         ! System Ready
        /*****************************************************************/
        /*                                                               */
        /*      Routine Initialization Per Hamby and Guillory           */
        /*                                                               */
        /*****************************************************************/
        D[1] = 0x55555555;                  ! Place Hex 55555555 Into D[1]
        A[0] = 0x1000;                      ! Place Hex 1000 Into A[0]
        A[1] = 0x2001;                      ! Place Illegal Address
        PC = 0x1000;                        ! Place Hex 1000 Into Program Counter
        M[0xc] = 0x0;                      ! Place Exception
        M[0xd] = 0x0;                      ! Vector Beginning In
        M[0xe] = 0x20;                     ! Location
        M[0xf] = 0x40;                     ! C Hex
        SA7 = 0x0786;                       ! Initialize System Stack Pointer
                                            ! At Program Headed Down (Added)
        next                                ! Execute Assignments
        )

/************************************************************************/
/*                                                                      */
/*  Initial Fetch Cycle. This cycle was not modeled but is necessary   */
/*  to retrieve modeled instructions for simulation and analysis. It   */
/*  was fashsioned from the Read Cycle described by Hamby and Guillory */
/*  on page VI-15 of their thesis.                                      */
/*                                                                      */
/************************************************************************/

fetch_initial_instruction :=
      (

      /******************************************************************/

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 0
      RW = hi;                              ! Memory Read
      ADENABLE = lo;                        ! Disable Address Bus Buffer
      DBENABLE = lo;                        ! Disable Data Bus Buffer
      IABUS = PC;                           ! Place PC On Internal Address
                                            ! Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2 Of
      PHI2 = hi;                            ! Clock Cycle 0
      ADENABLE = hi;                        ! Enable Address Bus Buffer
      EXABUF = IABUS;                       ! Gate Internal Address Bus
                                            ! Into External Address Buffer
      FCMODE = SRMODE;                      ! User Mode
      FCSPACE = 2;                          ! Accessing Program
      next;                                 ! Execute Impending Assignments
      ABUS = EXABUF;                        ! Address Placed On Bus(Added)
      next;                                 ! Execute Pending Assignments

      /******************************************************************/
      T = 1;                                ! Clock Cycle 1
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1 Of
      PHI2 = lo;                            ! Clock Cycle 1
      ASN = lo;                             ! Assert Address Strobe
      LDSN = lo;                            ! Assert Lower Data Strobe
      UDSN = lo;                            ! Assert Upper Data Strobe
      DBENABLE = hi;                        ! Enable Data Bus
      next;                                 ! Execute Pending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 1
      next;                                 ! Execute Pending Assignments

      /******************************************************************/
```

```
T = 2;                                        ! Clock Cycle 2
next;                                         ! Execute Assignment

PHI1 = hi;                                    ! Phase 1
PHI2 = lo;                                    ! Of Clock Cycle 2
while DTACKN eql hi                           ! Wait For Memory To Place
      (                                       ! Data On The Bus
      next;                                   ! Execute Impending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 2
      next;                                   ! Execute Assignments

      /*************************************************************/
      T = 3;                                  ! Clock Cycle 3
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];                   ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];                ! On Data Bus And
      DTACKN = lo;                            ! Asserts DTACKN(Added)
      next;                                   ! Execute Pending Assignments

      /*************************************************************/
      T = 2                                   ! Return To  Phase 2
                                              ! Of Clock Cycle 2
      );
      next;                                   ! Execute Impending Assignments

/*******************************************************************/
T = 3;                                        ! Clock Cycle 3
next;                                         ! Execute Assignment

PHI1 = lo;                                    ! Phase 2
PHI2 = hi;                                    ! Of Clock Cycle 3
EXDBUF = DBUS;                                ! Instruction On Data Bus
                                              ! Is Placed In External Data
                                              ! Bus Buffer
next;                                         ! Execute Pending Assignments

/*******************************************************************/
T = 4;                                        ! Clock Cycle 4
next;                                         ! Execute Assignment

PHI1 = hi;                                    ! Phase 1
PHI2 = lo;                                    ! Of Clock Cycle 4
PFR = EXDBUF;                                 ! The Contents Of The External
                                              ! Data Bus Buffer Are Placed
                                              ! In Prefetch Register
next;                                         ! Execute Pending Assignments

PHI1 = lo;                                    ! Phase 2
```

```
        PHI2 = hi;                          ! Of Clock Cycle 4
        ASN = hi;                           ! Deactivate Address Strobe
        LDSN = hi;                          ! Deactivate Lower Data Strobe
        UDSN = hi;                          ! Deactivate Upper Data Strobe
        IR = PFR;                           ! Contents Of Prefetch Register
                                            ! Are Placed Into Instruction
                                            ! Register
        DTACKN = hi;                        ! Deactivate Data Transfer(Added)
                                            ! Acknowledge
        PC = PC + 4;                        ! Increment Program Counter
        next;                               ! Execute Pending Assignments
        T = 0                               ! Reset Clock Cycle Counter
        )

   andi :=                                  ! AND.W #$DFFF,SR
       (
       SRMODE = lo;                         ! Effect Of Instruction
       IR<15:8> = M[PC];                    ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                ! Is To Set Status Register
          PC = PC + 2;                        ! Increment Program Counter
       T = 5;                               ! Supervisor Bit To User
       next;                                ! Mode
       T = 0                                ! Requires 6 Clock Cycles
       )

   addq :=                                  ! ADDQ.L #4,A7
       (
       if SRMODE                            ! Effect Of Instruction
          SA7 = SA7 + 4                     ! Is To Increment Either
       else                                 ! System Or User Stack
          UA7 = UA7 + 4;                    ! Register By 4 Depending On
       IR<15:8> = M[PC];                    ! Prefetch Next Instruction
       IR<7:0> = M[PC + 1];
       next;                                ! Is To Set Status Register
       PC = PC + 2;                         ! Increment Program Counter
       T = 7;                               ! Requires 8 Clock Cycles
       next;
       T = 0
       )

   illegal :=                               ! Illegal Address (0AFD)
          (

          /*****************************************************************/

        PHI1 = hi;                          ! Phase 1 Of
        PHI2 = lo;                          ! Clock Cycle 0
        RW = hi;                            ! Memory Read
        next;                               ! Execute Pending Assignments

        PHI1 = lo;                          ! Phase 2 Of
        PHI2 = hi;                          ! Clock Cycle 0
```

```
ADENABLE = lo;                              ! Disable Address Bus Buffer
DBENABLE = lo;                              ! Disable Data Bus Buffer
DBUS = 0xffff;                              ! Data Bus High Impedanced
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 1;                                      ! Clock Cycle 1
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1 Of
PHI2 = lo;                                  ! Clock Cycle 1
IDBUS = SR;                                 ! Place Status Register On
                                            ! Internal Data Bus
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 1
SRTEMP = IDBUS;                             ! Place Status Register
                                            ! On Bus In Temporary Register
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 2;                                      ! Clock Cycle 2
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 2
SRMODE = hi;                                ! Set Supervisor State
SRTRACE = lo;                               ! Turn Off Trace
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 4
next;                                       ! Execute Assignment

/******************************************************************/
T = 3;                                      ! Clock Cycle 3
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 3
SA7 = SA7 - 2;                              ! Decrement System Stack Pointer
                                            ! To Point To Location That Will
                                            ! Receive PC's Low Word
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 3
next;                                       ! Execute Pending Assignments

/******************************************************************/
T = 4;                                      ! Clock Cycle 4
next;
```

```
PHI1 = hi;                                      ! Phase 1 Of
PHI2 = lo;                                      ! Clock Cycle 4
VECADR = 3;                                     ! Place Vector Number In Register
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2 Of
PHI2 = hi;                                      ! Clock Cycle 4
next;                                           ! Execute Impending Assignments

/*****************************************************************/
T = 5;                                          ! Clock Cycle 5
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1 Of
PHI2 = lo;                                      ! Clock Cycle 5
VECADR = VECADR *:arith 2;                      ! Multiply Vector Number
                                                ! By 4 For Vector Address
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 5
next;                                           ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                          ! Clock Cycle 6
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 6
TEMPADR = EXABUF;                               ! Save Current Address
next;                                           ! In Temporary Register

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 6
next;

/*****************************************************************/
T = 7;                                          ! Clock Cycle 7
next;                                           ! Execute Assignment

PHI1 = hi;                                      ! Phase 1
PHI2 = lo;                                      ! Of Clock Cycle 7
next;                                           ! Execute Pending Assignments

PHI1 = lo;                                      ! Phase 2
PHI2 = hi;                                      ! Of Clock Cycle 7
next;

/*****************************************************************/
T = 8;                                          ! Clock Cycle 8
next;                                           ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 8
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
DBUS = 0xffff;                          ! Data Bus High Impedanced
IABUS = SA7;                             ! Place SA7 On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 8
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
IDBUS = PCLOW;                          ! Place Low Word from PC Onto
                                        ! Internal Data Bus
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 9;                                  ! Clock Cycle 9
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 9
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Place Internal Data Bus
                                        ! Contents Into External Data Buffer
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 9
DBUS = EXDBUF;                          ! Contents Of External Data Buffer
                                        ! Placed On Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 10;                                  ! Clock Cycle 10
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 10
UDSN = lo;                              ! Activate Upper Data Strobe
LDSN = lo;                               ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                     ! Wait For Memory To Place
```

```
      (                                    ! Data On The Bus
      twait = twait + 1;
      next;                                ! Execute Impending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 10
      next;                                ! Execute Assignments

      /*******************************************************/
      T = 11;                               ! Clock Cycle 11
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 11
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;                ! PC Low Word
      M[ABUS + 1] = DBUS<7:0>;              ! Stored
      DTACKN = lo                         ! Asserts DTACKN(Added)
      );
      next;                                ! Execute Pending Assignments

      /*******************************************************/
      T = 10                                ! Return To  Phase 2
                                           ! Of Clock Cycle 10
      );
      next;                                ! Execute Impending Assignments

  /**********************************************************/
  T = 11;                                   ! Clock Cycle 11
  next;                                    ! Execute Assignment

  PHI1 = lo;                               ! Phase 2
  PHI2 = hi;                               ! Of Clock Cycle 11
  SA7 = SA7 - 4;                           ! Set System Stack Pointer
                                           ! To Point To Status Register
                                           ! Storage Location
  next;                                    ! Execute Pending Assignments

  /**********************************************************/
  T = 12;                                   ! Clock Cycle 12
  next;                                    ! Execute Assignment

  PHI1 = hi;                               ! Phase 1
  PHI2 = lo;                               ! Of Clock Cycle 12
  next;                                    ! Execute Pending Assignments

  PHI1 = lo;                               ! Phase 2
  PHI2 = hi;                               ! Of Clock Cycle 12
  ASN = hi;                                ! Deactivate Address Strobe
  LDSN = hi;                               ! Deactivate Lower Data Strobe
  UDSN = hi;                               ! Deactivate Upper Data Strobe
  DTACKN = hi;                             ! Deactivate Data Transfer(Added)
```

```
                                         ! Acknowledge
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 13;                                  ! Clock Cycle 13
next;

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 13
RW = hi;                                 ! Memory Read
ADENABLE = lo;                           ! Disable Address Bus Buffer
DBENABLE = lo;                           ! Disable Data Bus Buffer
IABUS = SA7;                             ! Place SA7 On Internal Address
                                         ! Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2 Of
PHI2 = hi;                               ! Clock Cycle 13
ADENABLE = hi;                           ! Enable Address Bus Buffer
DBUS = 0xffff;                           ! Data Bus High Impedanced
EXABUF = IABUS;                          ! Gate Internal Address Bus
                                         ! Into External Address Buffer
FCMODE = SRMODE;                         ! Supervisor Mode
FCSPACE = 1;                             ! Accessing Data
IDBUS = SRTEMP;                          ! Place Holder Of Status Register
                                         ! Onto Internal Data Bus
next;                                    ! Execute Impending Assignments
ABUS = EXABUF;                           ! Address Placed On Bus(Added)
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 14;                                  ! Clock Cycle 14
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 14
ASN = lo;                                ! Assert Address Strobe
RW = lo;                                 ! Activate Write
EXDBUF = IDBUS;                          ! Internal Data Bus Moved
                                         ! To External Data Buffer
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 14
DBUS = EXDBUF;                           ! Contents Of External Data
                                         ! Buffer Placed On Data Bus
DBENABLE = hi;                           ! Enable Data Bus
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 15;                                  ! Clock Cycle 15
next;                                    ! Execute Assignment
```

```
PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 15
UDSN = lo;                                ! Activate Upper Data Strobe
LDSN = lo;                                 ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      twait = twait + 1;
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 15
      next;                               ! Execute Assignments

      /***********************************************************/
      T = 16;                              ! Clock Cycle 16
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 16
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;               ! PC Low Word
      M[ABUS + 1] = DBUS<7:0>;             ! Stored
      DTACKN = lo                        ! Asserts DTACKN(Added)
      );
      next;                               ! Execute Pending Assignments

      /***********************************************************/
      T = 15                               ! Return To  Phase 2
                                          ! Of Clock Cycle 15
      );
      next;                               ! Execute Impending Assignments

/*****************************************************************/
T = 16;                                    ! Clock Cycle 16
next;                                     ! Execute Assignment

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 16
SA7 = SA7 + 2;                            ! Set System Stack Pointer
                                          ! To Point To High PC
                                          ! Storage Location
next;                                     ! Execute Pending Assignments

/*****************************************************************/
T = 17;                                    ! Clock Cycle 17
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 17
next;                                     ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 17
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 18;                                 ! Clock Cycle 18
next;                                   ! Execute Pending Assignments

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 18
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = SA7;                            ! Place SA7 On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 18
ADENABLE = hi;                          ! Enable Address Bus Buffer
DBUS = 0xffff;                          ! Data Bus High Impedanced
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
IDBUS = PCHI;                           ! Place High Word Of PC
                                        ! Onto Internal Data Bus
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 19;                                  ! Clock Cycle 19
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 19
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Internal Data Bus Moved
                                        ! To External Data Buffer
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 19
DBUS = EXDBUF;                          ! Contents Of External Data
                                        ! Buffer Placed On Data Bus
```

```
DBENABLE = hi;                              ! Enable Data Bus
next;                                       ! Execute Pending Assignments

/*****************************************************************/
T = 20;                                      ! Clock Cycle 20
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 20
UDSN = lo;                                  ! Activate Upper Data Strobe
LDSN = lo;                                   ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      twait = twait + 1;
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 20
      next;                                 ! Execute Assignments

      /**********************************************************/
      T = 21;                                ! Clock Cycle 21
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 21
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;                 ! PC Low Word
      M[ABUS + 1] = DBUS<7:0>;               ! Stored
      DTACKN = lo                          ! Asserts DTACKN(Added)
      );
      next;                                 ! Execute Pending Assignments

      /**********************************************************/
      T = 20                                 ! Return To  Phase 2
                                            ! Of Clock Cycle 20
      );
      next;                                 ! Execute Impending Assignments

/*****************************************************************/
T = 21;                                      ! Clock Cycle 21
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 21
SA7 = SA7 - 4;                              ! Set System Stack Pointer
                                            ! To Point To Saved IR
next;                                       ! Execute Pending Assignments

/*****************************************************************/
```

```
T = 22;                                 ! Clock Cycle 22
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 22
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 22
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 23;                                 ! Clock Cycle 23
next;                                   ! Execute Pending Assignments

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 23
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = SA7;                            ! Place SA7 On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 23
ADENABLE = hi;                          ! Enable Address Bus Buffer
DBUS = 0xffff;                          ! Data Bus High Impedanced
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
IDBUS = IRTEMP;                         ! Place IR Causing Address Error
                                        ! Onto Internal Data Bus
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 24;                                 ! Clock Cycle 24
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 24
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Internal Data Bus Moved
                                        ! To External Data Buffer
```

```
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 24
DBUS = EXDBUF;                          ! Contents Of External Data
                                        ! Buffer Placed On Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/***************************************************************/
T = 25;                                  ! Clock Cycle 25
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 25
UDSN = lo;                              ! Activate Upper Data Strobe
LDSN = lo;                               ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      twait = twait + 1;
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 25
      next;                             ! Execute Assignments

      /*********************************************************/
      T = 26;                            ! Clock Cycle 26
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 26
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;             ! PC Low Word
      M[ABUS + 1] = DBUS<7:0>;           ! Stored
      DTACKN = lo                      ! Asserts DTACKN(Added)
      );
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 25                             ! Return To  Phase 2
                                        ! Of Clock Cycle 25
      );
      next;                             ! Execute Impending Assignments

/***************************************************************/
T = 26;                                  ! Clock Cycle 26
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
```

```
PHI2 = hi;                               ! Of Clock Cycle 26
SA7 = SA7 - 2;                           ! Set System Stack Pointer
                                         ! To Point To Saved Address
                                         ! Causing Exception (Low Word)
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 27;                                   ! Clock Cycle 27
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 27
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 27
ASN = hi;                                ! Deactivate Address Strobe
LDSN = hi;                               ! Deactivate Lower Data Strobe
UDSN = hi;                               ! Deactivate Upper Data Strobe
DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 28;                                  ! Clock Cycle 28
next;                                    ! Execute Pending Assignments

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 28
RW = hi;                                 ! Memory Read
ADENABLE = lo;                           ! Disable Address Bus Buffer
DBENABLE = lo;                           ! Disable Data Bus Buffer
IABUS = SA7;                             ! Place SA7 On Internal Address
                                         ! Bus
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2 Of
PHI2 = hi;                               ! Clock Cycle 28
ADENABLE = hi;                           ! Enable Address Bus Buffer
DBUS = 0xffff;                           ! Data Bus High Impedanced
EXABUF = IABUS;                          ! Gate Internal Address Bus
                                         ! Into External Address Buffer
FCMODE = SRMODE;                         ! Supervisor Mode
FCSPACE = 1;                             ! Accessing Data
IDBUS = TEMPADRLOW;                      ! Place Low Word Of TEMPADR
                                         ! Onto Internal Data Bus
next;                                    ! Execute Impending Assignments
ABUS = EXABUF;                           ! Address Placed On Bus(Added)
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 29;                                   ! Clock Cycle 29
next;                                    ! Execute Assignment
```

```
PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 29
ASN = lo;                               ! Assert Address Strobe
RW = lo;                                ! Activate Write
EXDBUF = IDBUS;                         ! Internal Data Bus Moved
                                        ! To External Data Buffer
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 29
DBUS = EXDBUF;                          ! Contents Of External Data
                                        ! Buffer Placed On Data Bus
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 30;                                  ! Clock Cycle 30
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 30
UDSN = lo;                              ! Activate Upper Data Strobe
LDSN = lo;                               ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                     ! Wait For Memory To Place
    (                                   ! Data On The Bus
    twait = twait + 1;
    next;                               ! Execute Impending Assignments

    PHI1 = lo;                          ! Phase 2
    PHI2 = hi;                          ! Of Clock Cycle 30
    next;                               ! Execute Assignments

    /*********************************************************/
    T = 31;                              ! Clock Cycle 31
    next;                               ! Execute Assignment

    PHI1 = hi;                          ! Phase 1
    PHI2 = lo;                          ! Of Clock Cycle 31
    if twait eql 2
    (
    M[ABUS] = DBUS<15:8>;               ! TEMPADR Low Word
    M[ABUS + 1] = DBUS<7:0>;             ! Stored
    DTACKN = lo                        ! Asserts DTACKN(Added)
    );
    next;                               ! Execute Pending Assignments

    /*********************************************************/
    T = 30                               ! Return To  Phase 2
                                        ! Of Clock Cycle 30
    );
```

```
          next;                                 ! Execute Impending Assignments

      /******************************************************************/
      T = 31;                                    ! Clock Cycle 31
      next;                                     ! Execute Assignment

      PHI1 = lo;                                ! Phase 2
      PHI2 = hi;                                ! Of Clock Cycle 31
      SA7 = SA7 - 4;                            ! Set System Stack Pointer
                                                ! To Point To Saved Access Type
      next;                                     ! Execute Pending Assignments

      /******************************************************************/
      T = 32;                                    ! Clock Cycle 32
      next;                                     ! Execute Assignment

      PHI1 = hi;                                ! Phase 1
      PHI2 = lo;                                ! Of Clock Cycle 32
      next;                                     ! Execute Pending Assignments

      PHI1 = lo;                                ! Phase 2
      PHI2 = hi;                                ! Of Clock Cycle 32
      ASN = hi;                                 ! Deactivate Address Strobe
      LDSN = hi;                                ! Deactivate Lower Data Strobe
      UDSN = hi;                                ! Deactivate Upper Data Strobe
      DTACKN = hi;                              ! Deactivate Data Transfer(Added)
                                                ! Acknowledge
      next;                                     ! Execute Pending Assignments

      /******************************************************************/
      T = 33;                                   ! Clock Cycle 33
      next;                                     ! Execute Pending Assignments

      PHI1 = hi;                                ! Phase 1 Of
      PHI2 = lo;                                ! Clock Cycle 33
      RW = hi;                                  ! Memory Read
      ADENABLE = lo;                            ! Disable Address Bus Buffer
      DBENABLE = lo;                            ! Disable Data Bus Buffer
      IABUS = SA7;                              ! Place SA7 On Internal Address
                                                ! Bus
      next;                                     ! Execute Pending Assignments

      PHI1 = lo;                                ! Phase 2 Of
      PHI2 = hi;                                ! Clock Cycle 33
      ADENABLE = hi;                            ! Enable Address Bus Buffer
      DBUS = 0xffff;                            ! Data Bus High Impedanced
      EXABUF = IABUS;                           ! Gate Internal Address Bus
                                                ! Into External Address Buffer
      FCMODE = SRMODE;                          ! Supervisor Mode
      FCSPACE = 1;                              ! Accessing Data
      IDBUS = ACTYPE;                           ! Place ACTYPE
                                                ! Onto Internal Data Bus
      next;                                     ! Execute Impending Assignments
```

```
ABUS = EXABUF;                            ! Address Placed On Bus(Added)
next;                                     ! Execute Pending Assignments

/***************************************************************/
T = 34;                                    ! Clock Cycle 34
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1 Of
PHI2 = lo;                                ! Clock Cycle 34
ASN = lo;                                 ! Assert Address Strobe
RW = lo;                                  ! Activate Write
EXDBUF = IDBUS;                           ! Internal Data Bus Moved
                                          ! To External Data Buffer
next;                                     ! Execute Pending Assignments

PHI1 = lo;                                ! Phase 2
PHI2 = hi;                                ! Of Clock Cycle 34
DBUS = EXDBUF;                            ! Contents Of External Data
                                          ! Buffer Placed On Data Bus
DBENABLE = hi;                            ! Enable Data Bus
next;                                     ! Execute Pending Assignments

/***************************************************************/
T = 35;                                    ! Clock Cycle 35
next;                                     ! Execute Assignment

PHI1 = hi;                                ! Phase 1
PHI2 = lo;                                ! Of Clock Cycle 35
UDSN = lo;                                ! Activate Upper Data Strobe
LDSN = lo;                                 ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                       ! Wait For Memory To Place
      (                                   ! Data On The Bus
      twait = twait + 1;
      next;                               ! Execute Impending Assignments

      PHI1 = lo;                          ! Phase 2
      PHI2 = hi;                          ! Of Clock Cycle 35
      next;                               ! Execute Assignments

      /*********************************************************/
      T = 36;                              ! Clock Cycle 36
      next;                               ! Execute Assignment

      PHI1 = hi;                          ! Phase 1
      PHI2 = lo;                          ! Of Clock Cycle 36
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;               ! ACTYPE
      M[ABUS + 1] = DBUS<7:0>;             ! Stored
      DTACKN = lo                        ! Asserts DTACKN(Added)
      );
```

```
          next;                               ! Execute Pending Assignments

          /*********************************************************/
          T = 35                               ! Return To  Phase 2
                                              ! Of Clock Cycle 35
          );
          next;                               ! Execute Impending Assignments

      /**************************************************************/
      T = 36;                                  ! Clock Cycle 36
      next;                                   ! Execute Assignment

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 36
      SA7 = SA7 + 2;                          ! Set System Stack Pointer
                                              ! To Point To Address Causing
                                              ! Exception (High Word)
      next;                                   ! Execute Pending Assignments

      /**************************************************************/
      T = 37;                                  ! Clock Cycle 37
      next;                                   ! Execute Assignment

      PHI1 = hi;                              ! Phase 1
      PHI2 = lo;                              ! Of Clock Cycle 37
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2
      PHI2 = hi;                              ! Of Clock Cycle 37
      ASN = hi;                               ! Deactivate Address Strobe
      LDSN = hi;                              ! Deactivate Lower Data Strobe
      UDSN = hi;                              ! Deactivate Upper Data Strobe
      DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                              ! Acknowledge
      next;                                   ! Execute Pending Assignments

      /**************************************************************/
      T = 38;                                 ! Clock Cycle 38
      next;                                   ! Execute Pending Assignments

      PHI1 = hi;                              ! Phase 1 Of
      PHI2 = lo;                              ! Clock Cycle 38
      RW = hi;                                ! Memory Read
      ADENABLE = lo;                          ! Disable Address Bus Buffer
      DBENABLE = lo;                          ! Disable Data Bus Buffer
      IABUS = SA7;                            ! Place SA7 On Internal Address
                                              ! Bus
      next;                                   ! Execute Pending Assignments

      PHI1 = lo;                              ! Phase 2 Of
      PHI2 = hi;                              ! Clock Cycle 38
      ADENABLE = hi;                          ! Enable Address Bus Buffer
      DBUS = 0xffff;                          ! Data Bus High Impedanced
```

```
EXABUF = IABUS;                          ! Gate Internal Address Bus
                                         ! Into External Address Buffer
FCMODE = SRMODE;                         ! Supervisor Mode
FCSPACE = 1;                             ! Accessing Data
IDBUS = TEMPADRHI;                       ! Place High Word Of TEMPADR
                                         ! Onto Internal Data Bus
next;                                    ! Execute Impending Assignments
ABUS = EXABUF;                           ! Address Placed On Bus(Added)
next;                                    ! Execute Pending Assignments

/****************************************************************/
T = 39;                                   ! Clock Cycle 39
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1 Of
PHI2 = lo;                               ! Clock Cycle 39
ASN = lo;                                ! Assert Address Strobe
RW = lo;                                 ! Activate Write
EXDBUF = IDBUS;                          ! Internal Data Bus Moved
                                         ! To External Data Buffer
next;                                    ! Execute Pending Assignments

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 39
DBUS = EXDBUF;                           ! Contents Of External Data
                                         ! Buffer Placed On Data Bus
DBENABLE = hi;                           ! Enable Data Bus
next;                                    ! Execute Pending Assignments

/****************************************************************/
T = 40;                                   ! Clock Cycle 40
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 40
UDSN = lo;                               ! Activate Upper Data Strobe
LDSN = lo;                                ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      twait = twait + 1;
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 40
      next;                              ! Execute Assignments

      /*********************************************************/
      T = 41;                             ! Clock Cycle 41
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 41
      if twait eql 2
      (
      M[ABUS] = DBUS<15:8>;             ! TEMPADR High Word
      M[ABUS + 1] = DBUS<7:0>;           ! Stored
      DTACKN = lo                      ! Asserts DTACKN(Added)
      );
      next;                             ! Execute Pending Assignments

      /*********************************************************/
      T = 40                             ! Return To  Phase 2
                                        ! Of Clock Cycle 40
      );
      next;                             ! Execute Impending Assignments

/************************************************************/
T = 41;                                  ! Clock Cycle 41
next;                                   ! Execute Assignment

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 41
next;                                   ! Execute Pending Assignments

/************************************************************/
T = 42;                                   ! Clock Cycle 42
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 42
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 42
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
next;                                   ! Execute Pending Assignments

/************************************************************/

T = 43;                                  ! Clock Cycle 43
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 43
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = VECADR;                         ! Place VECADR On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments
```

```
PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 43
ADENABLE = hi;                          ! Enable Address Bus Buffer
DBUS = 0xffff;                          ! Data Bus High Impedanced
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 44;                                  ! Clock Cycle 44
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 44
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 44
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 45;                                  ! Clock Cycle 45
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 45
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 45
      next;                             ! Execute Assignments

      /*******************************************************/
      T = 46;                            ! Clock Cycle 46
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 46
      DBUS<15:8> = M[ABUS];             ! Memory Places address
      DBUS<7:0> = M[ABUS + 1];          ! On Data Bus And
      DTACKN = lo;                      ! Asserts DTACKN(Added)
      next;                             ! Execute Pending Assignments
```

```
        /**************************************************************/
        T = 45;                               ! Return To  Phase 2
                                             ! Of Clock Cycle 45
        );
        next;                                ! Execute Impending Assignments

    /**************************************************************/
    T = 46;                                   ! Clock Cycle 46
    next;                                    ! Execute Assignment

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 46
    EXDBUF = DBUS;                           ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
    next;                                    ! Execute Pending Assignments

    /**************************************************************/
    T = 47;                                   ! Clock Cycle 47
    next;                                    ! Execute Assignment

    PHI1 = hi;                               ! Phase 1
    PHI2 = lo;                               ! Of Clock Cycle 47
    HANADRLOW = EXDBUF;                      ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Handler Routine Low Address
    next;                                    ! Execute Pending Assignments

    PHI1 = lo;                               ! Phase 2
    PHI2 = hi;                               ! Of Clock Cycle 47
    ASN = hi;                                ! Deactivate Address Strobe
    LDSN = hi;                               ! Deactivate Lower Data Strobe
    UDSN = hi;                               ! Deactivate Upper Data Strobe
    DTACKN = hi;                             ! Deactivate Data Transfer(Added)
                                             ! Acknowledge
    VECADR = VECADR + 2;                     ! Increment Vector Address Register
                                             ! To Pick Handler Address Low Word
    next;                                    ! Execute Pending Assignments

    /**************************************************************/

    T = 48;                                   ! Clock Cycle 48
    next;                                    ! Execute Assignment

    PHI1 = hi;                               ! Phase 1
    PHI2 = lo;                               ! Of Clock Cycle 48
    RW = hi;                                 ! Memory Read
    ADENABLE = lo;                           ! Disable Address Bus Buffer
    DBENABLE = lo;                           ! Disable Data Bus Buffer
    DBUS = 0xffff;                           ! Data Bus High Impedanced
    IABUS = VECADR;                          ! Place VECADR On Internal Address
                                             ! Bus
```

```
IDBUS = HANADRLOW;                      ! Move Handler High Address To
                                        ! Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 48
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! Supervisor Mode
FCSPACE = 1;                            ! Accessing Data
HANADRHI = IDBUS;                       ! Move Handler High Address
                                        ! To Upper Word Of Register
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 49;                                  ! Clock Cycle 49
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 49
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 49
next;                                   ! Execute Pending Assignments

/*************************************************************/
T = 50;                                  ! Clock Cycle 50
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 50
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 50
      next;                             ! Execute Assignments

      /*******************************************************/
      T = 51;                            ' Clock Cycle 51
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 51
```

```
          DBUS<15:8> = M[ABUS];              ! Memory Places Address
          DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
          DTACKN = lo;                       ! Asserts DTACKN(Added)
          next;                              ! Execute Pending Assignments

          /****************************************************/
          T = 50;                             ! Return To  Phase 2
                                             ! Of Clock Cycle 50
          );
          next;                              ! Execute Impending Assignments

     /*********************************************************/
     T = 51;                                  ! Clock Cycle 51
     next;                                   ! Execute Assignment

     PHI1 = lo;                              ! Phase 2
     PHI2 = hi;                              ! Of Clock Cycle 51
     EXDBUF = DBUS;                          ! Instruction On Data Bus
                                             ! Is Placed In External Data
                                             ! Bus Buffer
     next;                                   ! Execute Pending Assignments

     /*********************************************************/
     T = 52;                                  ! Clock Cycle 52
     next;                                   ! Execute Assignment

     PHI1 = hi;                              ! Phase 1
     PHI2 = lo;                              ! Of Clock Cycle 52
     HANADRLOW = EXDBUF;                     ! The Contents Of The External
                                             ! Data Bus Buffer Are Placed
                                             ! In Handler Routine Low Address
     next;                                   ! Execute Pending Assignments

     PHI1 = lo;                              ! Phase 2
     PHI2 = hi;                              ! Of Clock Cycle 52
     ASN = hi;                               ! Deactivate Address Strobe
     LDSN = hi;                              ! Deactivate Lower Data Strobe
     UDSN = hi;                              ! Deactivate Upper Data Strobe
     DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                             ! Acknowledge
     next;                                   ! Execute Pending Assignments

     /*********************************************************/

     T = 53;                                  ! Clock Cycle 53
     next;                                   ! Execute Assignment

     PHI1 = hi;                              ! Phase 1 Of
     PHI2 = lo;                              ! Clock Cycle 53
     RW = hi;                                ! Memory Read
     ADENABLE = lo;                          ! Disable Address Bus Buffer
     DBENABLE = lo;                          ! Disable Data Bus Buffer
     DBUS = 0xffff;                          ! Data Bus High Impedanced
```

```
IABUS = HANADR;                         ! Place HANADR On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 53
ADENABLE = hi;                          ! Enable Address Bus Buffer
EXABUF = IABUS;                         ! Gate Internal Address Bus
                                        ! Into External Address Buffer
FCMODE = SRMODE;                        ! User Mode
PC = IABUS;                             ! Place HANADR In PC
FCSPACE = 2;                            ! Accessing Program
next;                                   ! Execute Impending Assignments
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 54;                                  ! Clock Cycle 54
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 54
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 54
next;                                   ! Execute Pending Assignments

/****************************************************************/
T = 55;                                  ! Clock Cycle 55
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1
PHI2 = lo;                              ! Of Clock Cycle 55
while DTACKN eql hi                     ! Wait For Memory To Place
      (                                 ! Data On The Bus
      next;                             ! Execute Impending Assignments

      PHI1 = lo;                        ! Phase 2
      PHI2 = hi;                        ! Of Clock Cycle 55
      next;                             ! Execute Assignments

      /**********************************************************/
      T = 56;                            ! Clock Cycle 56
      next;                             ! Execute Assignment

      PHI1 = hi;                        ! Phase 1
      PHI2 = lo;                        ! Of Clock Cycle 56
      DBUS<15:8> = M[ABUS];             ! Memory Places Instruction
```

```
          DBUS<7:0> = M[ABUS + 1];         ! On Data Bus And
          DTACKN = lo;                     ! Asserts DTACKN(Added)
          next;                            ! Execute Pending Assignments

          /*****************************************************/
          T = 55;                           ! Return To  Phase 2
                                           ! Of Clock Cycle 55
          );
          next;                            ! Execute Impending Assignments

      /************************************************************/
      T = 56;                               ! Clock Cycle 56
      next;                                ! Execute Assignment

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 56
      EXDBUF = DBUS;                       ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
      next;                                ! Execute Pending Assignments

      /************************************************************/
      T = 57;                               ! Clock Cycle 57
      next;                                ! Execute Assignment

      PHI1 = hi;                           ! Phase 1
      PHI2 = lo;                           ! Of Clock Cycle 57
      PFR = EXDBUF;                        ! The Contents Of The External
                                           ! Data Bus Buffer Are Placed
                                           ! In Prefetch Register
      next;                                ! Execute Pending Assignments

      PHI1 = lo;                           ! Phase 2
      PHI2 = hi;                           ! Of Clock Cycle 57
      ASN = hi;                            ! Deactivate Address Strobe
      LDSN = hi;                           ! Deactivate Lower Data Strobe
      UDSN = hi;                           ! Deactivate Upper Data Strobe
      IR = PFR;                            ! Contents Of Prefetch Register
                                           ! Are Placed Into Instruction
                                           ! Register
      DTACKN = hi;                         ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
      PC = PC + 2;                         ! Increment Program Counter
      next;                                ! Execute Pending Assignments

      /************************************************************/

      T = 58;                              ! Clock Cycle 58
      next;

      PHI1 = hi;                           ! Phase 1 Of
      PHI2 = lo;                           ! Clock Cycle 58
      ADENABLE = lo;                       ! Disable Address Bus Buffer
```

```
        DBENABLE = lo;                        ! Disable Data Bus Buffer
        DBUS = 0xffff;                        ! Data Bus High Impedanced
        ASN = hi;                             ! Disable Address,
        LDSN = hi;                            ! Lower Data, and
        UDSN = hi;                            ! Upper Data Strobes
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2 Of
        PHI2 = hi;                            ! Clock Cycle 58
        next;                                 ! Execute Impending Assignments

        /*****************************************************************/
        T = 59;                                ! Clock Cycle 59
        next;                                 ! Execute Assignment

        PHI1 = hi;                            ! Phase 1 Of
        PHI2 = lo;                            ! Clock Cycle 59
        next;                                 ! Execute Pending Assignments

        PHI1 = lo;                            ! Phase 2
        PHI2 = hi;                            ! Of Clock Cycle 59
        next;

        T = 0                                 ! Reset Clock Cycle Counter
        )

    rte :=                                     ! RTE (Return From Exception)
       (
       SA7 = SA7 - 2;                          ! Effect Of This Instruction
       next;                                   ! Is To Pop The PC And SR
       SR<15:8> = M[SA7];                      ! From The Stack
       SR<7:0> = M[SA7 + 1];
       next;
       SA7 = SA7 + 2;
       next;
       PC<31:24> = M[SA7];
       PC<23:16> = M[SA7 + 1];
       next;
       SA7 = SA7 + 2;
       next;
       PC<15:8> = M[SA7];
       PC<7:0> = M[SA7 + 1];
       next;
       IR<15:8> = M[PC];
       IR<7:0> = M[PC + 1];
       next;
       PC = PC + 2;
       next;
       T = 19;                                 ! Requires 20 Clock Cycles
       next;
       T = 0
       )
```

```
move :=                                         ! MOVE.W D1,(A1)
     (

     /*****************************************************************/

     PHI1 = hi;                                 ! Phase 1 Of
     PHI2 = lo;                                 ! Clock Cycle 0
     DBUS = 0xffff;                             ! Data Bus High Impedanced
     RW = hi;                                   ! Memory Read
     ADENABLE = lo;                             ! Disable Address Bus Buffer
     DBENABLE = lo;                             ! Disable Data Bus Buffer
     IABUS = PC;                                ! Place PC On Internal Address
                                                ! Bus
     next;                                      ! Execute Pending Assignments

     PHI1 = lo;                                 ! Phase 2 Of
     PHI2 = hi;                                 ! Clock Cycle 0
     ADENABLE = hi;                             ! Enable Address Bus Buffer
     EXABUF = IABUS;                            ! Gate Internal Address Bus
                                                ! Into External Address Buffer
     FCMODE = SRMODE;                           ! User Mode
     FCSPACE = 2;                               ! Accessing Program
     next;                                      ! Execute Impending Assignments
     ABUS = EXABUF;                             ! Address Placed On Bus(Added)
     next;                                      ! Execute Pending Assignments

     /*****************************************************************/
     T = 1;                                     ! Clock Cycle 1
     next;                                      ! Execute Assignment

     PHI1 = hi;                                 ! Phase 1 Of
     PHI2 = lo;                                 ! Clock Cycle 1
     ASN = lo;                                  ! Assert Address Strobe
     LDSN = lo;                                 ! Assert Lower Data Strobe
     UDSN = lo;                                 ! Assert Upper Data Strobe
     DBENABLE = hi;                             ! Enable Data Bus
     next;                                      ! Execute Pending Assignments

     PHI1 = lo;                                 ! Phase 2
     PHI2 = hi;                                 ! Of Clock Cycle 1
     next;                                      ! Execute Pending Assignments

     /*****************************************************************/
     T = 2;                                     ! Clock Cycle 2
     next;                                      ! Execute Assignment

     PHI1 = hi;                                 ! Phase 1
     PHI2 = lo;                                 ! Of Clock Cycle 2
     while DTACKN eql hi                        ! Wait For Memory To Place
           (                                    ! Data On The Bus
           next;                                ! Execute Impending Assignments

           PHI1 = lo;                           ! Phase 2
```

```
        PHI2 = hi;                          ! Of Clock Cycle 2
        next;                               ! Execute Assignments

        /*************************************************************/
        T = 3;                              ! Clock Cycle 3
        next;                               ! Execute Assignment

        PHI1 = hi;                          ! Phase 1
        PHI2 = lo;                          ! Of Clock Cycle 3
        DBUS<15:8> = M[ABUS];               ! Memory Places Instruction
        DBUS<7:0> = M[ABUS + 1];            ! On Data Bus And
        DTACKN = lo;                        ! Asserts DTACKN(Added)
        next;                               ! Execute Pending Assignments

        /*************************************************************/
        T = 2                               ! Return To  Phase 2
                                            ! Of Clock Cycle 2
        );
        next;                               ! Execute Impending Assignments

    /*****************************************************************/
    T = 3;                                  ! Clock Cycle 3
    next;                                   ! Execute Assignment

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 3
    EXDBUF = DBUS;                          ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
    next;                                   ! Execute Pending Assignments

    /*****************************************************************/
    T = 4;                                  ! Clock Cycle 4
    next;                                   ! Execute Assignment

    PHI1 = hi;                              ! Phase 1
    PHI2 = lo;                              ! Of Clock Cycle 4
    PFR = EXDBUF;                           ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
    next;                                   ! Execute Pending Assignments

    PHI1 = lo;                              ! Phase 2
    PHI2 = hi;                              ! Of Clock Cycle 4
    ASN = hi;                               ! Deactivate Address Strobe
    LDSN = hi;                              ! Deactivate Lower Data Strobe
    UDSN = hi;                              ! Deactivate Upper Data Strobe
                                            ! Are Placed Into Instruction
                                            ! Register
    PC = PC + 2;                            ! Increment Program Counter
    DTACKN = hi;                            ! Deactivate Data Transfer(Added)
                                            ! Acknowledge
    next;
```

```
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBUS = 0xffff;                          ! Data Bus Returned To High
                                        ! Impedance State
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = A[1];                           ! Place A[1] On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 1;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
IDBUS = D1LWORD;                        ! Place Low Word from D[1] On
                                        ! Internal Data Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
RW = lo;
EXDBUF = IDBUS;                         ! Place Contents Of Internal
                                        ! Data Bus Into External Data Buffer
SRCARRY = lo;                           ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SRNEG = lo;
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
DBUS = EXDBUF;                          ! Place Data On External Data Bus
DBENABLE = hi;                          ! Enable Data Bus
case ABUS<1>                            ! This Instruction Will Abort
     0:
      (
if EXDBUF eql 0                         ! Set Zero Condition Bit If Needed
   SRZERO = hi;
```

```
next;                                          ! Execute Pending Assignments

/**********************************************************************/
T = 7;                                         ! Clock Cycle 7
next;                                          ! Execute Assignment

PHI1 = hi;                                     ! Phase 1
PHI2 = lo;                                     ! Of Clock Cycle 7
if EXDBUF<15>                                  ! Set Negative Condition Bit
   SRNEG = hi;                                 ! If Needed
UDSN = lo;                                     ! Activate Upper And
LDSN = lo;                                     ! Lower Data Strobes
twait = 0;                                     ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi                            ! Wait For Memory To Place
      (                                        ! Data On The Bus
      twait = twait + 1;                       ! Increment Wait Cycle
      next;                                    ! Execute Impending Assignments

      PHI1 = lo;                               ! Phase 2
      PHI2 = hi;                               ! Of Clock Cycle 7
      next;                                    ! Execute Assignments

      /***************************************************************/
      T = 8;                                   ! Clock Cycle 8
      next;                                    ! Execute Assignment

      PHI1 = hi;                               ! Phase 1
      PHI2 = lo;                               ! Of Clock Cycle 8
      if twait eql 2                           ! Memory Responds After 2 Cycles
      (
      M[ABUS] = DBUS<15:8>;                    ! Store Data From Bus
      M[ABUS + 1] = DBUS<7:0>;                 ! In Memory
      DTACKN = lo                             ! Asserts DTACKN(Added)
      );
      next;                                    ! Execute Pending Assignments

      /***************************************************************/
      T = 7                                   ! Return To Phase 2
                                              ! Of Clock Cycle 7
      );
      next;                                   ! Execute Impending Assignments

/*******************************************************************
T = 8;                                         ! Clock Cycle 8
next;                                          ! Execute Assignment

PHI1 = lo;                                     ! Phase 2
PHI2 = hi;                                     ! Of Clock Cycle 8
next;                                          ! Execute Pending Assignments

/**********************************************************************/
T = 9;                                         ! Clock Cycle 9
```

```
next;                                        ! Execute Assignment

PHI1 = hi;                                   ! Phase 1
PHI2 = lo;                                   ! Of Clock Cycle 9
next;                                        ! Execute Pending Assignments

PHI1 = lo;                                   ! Phase 2
PHI2 = hi;                                   ! Of Clock Cycle 9
ASN = hi;                                    ! Deactivate Address Strobe
LDSN = hi;                                   ! Deactivate Lower Data Strobe
UDSN = hi;                                   ! Deactivate Upper Data Strobe
PC = PC + 2;                                 ! Increment Program Counter
IR = PFR;                                    ! Place Contents Of Prefetch
                                             ! Register Into Instruction
                                             ! Register
DTACKN = hi;                                 ! Deactivate Data Transfer
                                             ! Acknowledge(Added)
next                                        ! Execute Pending Assignments
  )
1:
  (                                            ! Terminate MOVE.W D1,(A1)
next;                                        ! Instruction Because Of Illegal

T = 7;                                       ! Address And Initiate Exception
next;                                        ! Processing

PHI1 = hi;
PHI2 = lo;
UDSN = lo;
LDSN = lo;
next;

PHI1 = lo;
PHI2 = hi;
IRTEMP = IR;                                 ! Instruction Register And
ACTYPE<2:0> = FC;                            ! User/System Data/Program
EXCEPT = hi;                                ! Status Saved In Temporary Registers
ACTYPE<4> = RW;                             ! Exception Occurred During
next;                                       ! Write Cycle

T = 8;                                         ! Clock Cycle 8
next;

PHI1 = hi;
PHI2 = lo;
ACTYPE<3> = lo;                          ! Instruction Caused Exception(Changed)
next;

PHI1 = lo;
PHI2 = hi;
ASN = hi;
UDSN = hi;
IR = 0x0afd;                             ! Illegal Address Exception
```

```
        LDSN = hi;
        next                                    ! Exception Processing

        )
        esac;
        T = 0
  )

  jmp :=                                        ! JMP (A0)
        (

        /*****************************************************************/

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 0
        DBUS = 0xffff;                          ! Place Data Bus In A High
                                                ! Impedance State (Added)
        RW = hi;                                ! Memory Read
        ADENABLE = lo;                          ! Disable Address Bus Buffer
        DBENABLE = lo;                          ! Disable Data Bus Buffer
        IABUS = PC;                             ! Place PC On Internal Address
                                                ! Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2 Of
        PHI2 = hi;                              ! Clock Cycle 0
        ADENABLE = hi;                          ! Enable Address Bus Buffer
        EXABUF = IABUS;                         ! Gate Internal Address Bus
                                                ! Into External Address Buffer
        FCMODE = SRMODE;                        ! User Mode
        FCSPACE = 2;                            ! Accessing Program
        next;                                   ! Execute Pending Assignments
        ABUS = EXABUF;                          ! Address Placed On Bus(Added)
        next;                                   ! Execute Pending Assignments

        /*****************************************************************/
        T = 1;                                  ! Clock Cycle 1
        next;                                   ! Execute Assignment

        PHI1 = hi;                              ! Phase 1 Of
        PHI2 = lo;                              ! Clock Cycle 1
        ASN = lo;                               ! Assert Address Strobe
        LDSN = lo;                              ! Assert Lower Data Strobe
        UDSN = lo;                              ! Assert Upper Data Strobe
        IABUS = A[0];                           ! Move Jump Address From A[0]
                                                ! To Internal Address Buffer
        DBENABLE = hi;                          ! Enable Data Bus
        next;                                   ! Execute Pending Assignments

        PHI1 = lo;                              ! Phase 2
        PHI2 = hi;                              ! Of Clock Cycle 1
        PC = IABUS;                             ! Place Jump Address Into Program
                                                ! Counter
```

```
next;

/*****************************************************************/
T = 2;                                   ! Clock Cycle 2
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 2
while DTACKN eql hi                      ! Wait For Memory To Place
      (                                  ! Data On The Bus
      next;                              ! Execute Impending Assignments

      PHI1 = lo;                         ! Phase 2
      PHI2 = hi;                         ! Of Clock Cycle 2
      next;                              ! Execute Assignments

      /***********************************************************/
      T = 3;                             ! Clock Cycle 3
      next;                              ! Execute Assignment

      PHI1 = hi;                         ! Phase 1
      PHI2 = lo;                         ! Of Clock Cycle 3
      DBUS<15:8> = M[ABUS];              ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];           ! On Data Bus And
      DTACKN = lo;                       ! Asserts DTACKN(Added)
      next;                              ! Execute Pending Assignments

      /***********************************************************/
      T = 2                              ! Return To Phase 2
                                         ! Of Clock Cycle 2
      );
      next;                              ! Execute Impending Assignments

/*****************************************************************/
T = 3;                                   ! Clock Cycle 3
next;                                    ! Execute Assignment

PHI1 = lo;                               ! Phase 2
PHI2 = hi;                               ! Of Clock Cycle 3
EXDBUF = DBUS;                           ! Instruction On Data Bus
                                         ! Is Placed In External Data
                                         ! Bus Buffer
next;                                    ! Execute Pending Assignments

/*****************************************************************/
T = 4;                                   ! Clock Cycle 4
next;                                    ! Execute Assignment

PHI1 = hi;                               ! Phase 1
PHI2 = lo;                               ! Of Clock Cycle 4
next;
PFR = EXDBUF;                            ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
```

```
                                        ! In Prefetch Register
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 4
ASN = hi;                               ! Deactivate Address Strobe
LDSN = hi;                              ! Deactivate Lower Data Strobe
UDSN = hi;                              ! Deactivate Upper Data Strobe
DTACKN = hi;                            ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
next;
/*****************************************************************/
T = 5;                                  ! Clock Cycle 5
next;                                   ! Execute Previous Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 5
RW = hi;                                ! Memory Read
ADENABLE = lo;                          ! Disable Address Bus Buffer
DBENABLE = lo;                          ! Disable Data Bus Buffer
IABUS = PC;                             ! Place PC On Internal Address
                                        ! Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2 Of
PHI2 = hi;                              ! Clock Cycle 5
ADENABLE = hi;                          ! Enable Address Bus Buffer
FCMODE = SRMODE;                        ! User Mode
FCSPACE = 2;                            ! Accessing Program
EXABUF = IABUS;                         ! Gate Internal Address Bus
next;                                   ! Into External Address Buffer
ABUS = EXABUF;                          ! Address Placed On Bus(Added)
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 6;                                  ! Clock Cycle 6
next;                                   ! Execute Assignment

PHI1 = hi;                              ! Phase 1 Of
PHI2 = lo;                              ! Clock Cycle 6
ASN = lo;                               ! Assert Address Strobe
LDSN = lo;                              ! Assert Lower Data Strobe
UDSN = lo;                              ! Assert Upper Data Strobe
DBENABLE = hi;                          ! Enable Data Bus
next;                                   ! Execute Pending Assignments

PHI1 = lo;                              ! Phase 2
PHI2 = hi;                              ! Of Clock Cycle 6
next;                                   ! Execute Pending Assignments

/*****************************************************************/
T = 7;                                  ! Clock Cycle 7
next;                                   ! Execute Assignment
```

```
PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 7
while DTACKN eql hi                         ! Wait For Memory To Place
      (                                     ! Data On The Bus
      next;                                 ! Execute Impending Assignments

      PHI1 = lo;                            ! Phase 2
      PHI2 = hi;                            ! Of Clock Cycle 7
      next;                                 ! Execute Assignments

      /*********************************************************/
      T = 8;                                ! Clock Cycle 8
      next;                                 ! Execute Assignment

      PHI1 = hi;                            ! Phase 1
      PHI2 = lo;                            ! Of Clock Cycle 8
      DBUS<15:8> = M[ABUS];                 ! Memory Places Instruction
      DBUS<7:0> = M[ABUS + 1];              ! On Data Bus And
      DTACKN = lo;                          ! Asserts DTACKN(Added)
      next;                                 ! Execute Pending Assignments

      /*********************************************************/
      T = 7                                ! Return To Phase 2
                                           ! Of Clock Cycle 7
      );
      next;                                ! Execute Impending Assignments

/*************************************************************/
T = 8;                                      ! Clock Cycle 8
next;                                       ! Execute Assignment

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 8
EXDBUF = DBUS;                              ! Instruction On Data Bus
                                            ! Is Placed In External Data
                                            ! Bus Buffer
next;                                       ! Execute Pending Assignments

/*************************************************************/
T = 9;                                      ! Clock Cycle 9
next;                                       ! Execute Assignment

PHI1 = hi;                                  ! Phase 1
PHI2 = lo;                                  ! Of Clock Cycle 9
PFR = EXDBUF;                               ! The Contents Of The External
                                            ! Data Bus Buffer Are Placed
                                            ! In Prefetch Register
next;                                       ! Execute Pending Assignments

PHI1 = lo;                                  ! Phase 2
PHI2 = hi;                                  ! Of Clock Cycle 9
ASN = hi;                                   ! Deactivate Address Strobe
```

```
        LDSN = hi;                                     ! Deactivate Lower Data Strobe
        UDSN = hi;                                     ! Deactivate Upper Data Strobe
        PC = PC + 4;                                   ! Increment Program Counter
        IR = PFR;                                      ! Place Contents Of Prefetch
                                                       ! Register Into Instruction
                                                       ! Register
        DTACKN = hi;                                   ! Deactivate Data Transfer
                                                       ! Acknowledge(Added)
        next;                                          ! Execute Pending Assignments
        T = 0                                          ! Reset Clock Cycle Counter
        )

    decode_execute_prefetch :=
                        (
                        case IR
                             0x3281: move    ! MOVE.W D1,(A1) with illegal addr
                             0x027c: andi    ! AND.W #$DFFF,SR
                             0x588f: addq    ! ADDQ.L #4,A7
                             047320: jmp     ! JMP (A0)
                             0x4e73: rte     ! RTE (Return From Exception)
                             0x0afd: illegal! Illegal Address Exception
                        esac
                        )

    main :=
        (
        power_on_initialize;
        fetch_initial_instruction;
        while READY eql hi
              (
              decode_execute_prefetch
              )
        )
```

Appendix D: MC68000 Metamicro Description

```
!*********************************************************!
!*                                                       *!
!* m68000.m.                                             *!
!* metaMicro description file for Motorola 68000         *!
!* microprocessor.                                       *!
!* This file is included in the first line of            *!
!* assembler source code file. say source.m.             *!
!* It generates output file source.n if 'micro' is       *!
!* used. If 'mas' is used than it generates the          *!
!* following output files.                               *!
!* source.n : nodal output file.                         *!
!* source.l : assembler listing, logical addresses.      *!
!* source.L : assembler listing, both logical and        *!
!*            physical addresses and assembled           *!
!*            object code listings.                      *!
!* l.out    : assmbled object code core image.           *!
!* Use 'micro' with 'cater' and 'merge' or use 'mas'.*!
!*                                                       *!
!* Author   : Samir S. Shah.                             *!
!* Date     : Fall 1979.                                 *!
!* Modified : Samir S. Shah.                             *!
!* Date     : Spring 1981.                               *!
!*                                                       *!
!*********************************************************!

!*********************************************************!
!*                                                       *!
!* Declaration for instruction lengths and widths.       *!
!* Memory is byte addressable,default instruction        *!
!* length is 2 bytes,maximum instruction length is       *!
!* 10 bytes.                                             *!
!*                                                       *!
!*********************************************************!

instr
  I[10,2]<8>$


format

!*********************************************************!
!*                                                       *!
!* Fields of instruction bytes zero and one.             *!
!*                                                       *!
!*********************************************************!

  Opcode     = I[0]<7:4>,
  Condition = I[0]<3:0>,
  Dst_Rn    = I[0]<3:1>,
  Dst_Mode0 = I[0]<0>,
  Aux_Op    = I[0]<0>,
  Dst_Mode1 = I[1]<7:6>,
  Size      = I[1]<7:6>,
  Src_Mode  = I[1]<5:3>,
  I_R       = I[1]<5>,
  S_R       = I[1]<4>,
  R_M       = I[1]<3>,
  Src_Rn    = I[1]<2:0>,

!****************************************************!
!*                                                  *!
!* Fields of instruction bytes Two and three.       *!
!*                                                  *!
!****************************************************!

  T_Ind  = I[2]<7>,
  T_Rn   = I[2]<6:4>,
  T_Size = I[2]<3>,
  T_Disp = I[3]<7:0>,

!****************************************************!
!*                                                  *!
!* Fields of instruction bytes Four and five.       *!
!*                                                  *!
!****************************************************!

  F_Ind  = I[4]<7>,
  F_Rn   = I[4]<6:4>,
  F_Size = I[4]<3>,
  F_Disp = I[5]<7:0>,

!****************************************************!
!*                                                  *!
!* Fields of instruction bytes Six and seven.       *!
!*                                                  *!
!****************************************************!

  S_Ind  = I[6]<7>,
  S_Rn   = I[6]<6:4>,
  S_Size = I[6]<3>,
  S_Disp = I[7]<7:0>,

!****************************************************!
!*                                                  *!
!* Shorter names for instruction bytes.             *!
!*                                                  *!
!****************************************************!

  I0 = I[0]<7:0>,
  I1 = I[1]<7:0>,
  I2 = I[2]<7:0>,
  I3 = I[3]<7:0>,
  I4 = I[4]<7:0>,
  I5 = I[5]<7:0>,
  I6 = I[6]<7:0>,
  I7 = I[7]<7:0>,
  I8 = I[8]<7:0>,
  I9 = I[9]<7:0>$


macro

!***********************************************************!
!*                                                         *!
!* Data registers.                                         *!
!*                                                         *!
!***********************************************************!

  D0 = 0 &,
  D1 = 1 &,
  D2 = 2 &,
  D3 = 3 &,
  D4 = 4 &,
  D5 = 5 &,
  D6 = 6 &,
  D7 = 7 &,

!***********************************************************!
!*                                                         *!
!* Address registers.                                      *!
!*                                                         *!
!***********************************************************!

  A0 = 0 &,
  A1 = 1 &,
  A2 = 2 &,
  A3 = 3 &,
  A4 = 4 &,
  A5 = 5 &,
  A6 = 6 &,
  A7 = 7 &,
  SP = 7 &,

!***********************************************************!
!*                                                         *!
!* Size.                                                   *!
!*                                                         *!
!***********************************************************!

  B = 0 &,
  W = 1 &,
  L = 2 &,
!***************************************************************!
!*                                                             *!
!* Addressing modes.                                           *!
!*                                                             *!
!***************************************************************!

  DR = 0 &,      ! Data Register direct
  AR = 1 &,      ! Address Register direct
  IR = 2 &,      ! Indirect Register
  AI = 3 &,      ! Auto-Increment
  AD = 4 &,      ! Auto-Decrement
  DS = 5 &,      ! DiSplacement
  IX = 6 &,      ! IndeX
  SP = 7 &,      ! SPecial
     SH = 0 &,   ! SHort
     LN = 1 &,   ! LoNg
     PD = 2 &,   ! Program counter Displacement
     PX = 3 &,   ! Program counter indeX
     IM = 4 &,   ! IMmediate

!***************************************************************!
!*                                                             *!
!* Register or Memory mode selection.                          *!
!*                                                             *!
!***************************************************************!

  R = 0 &,
  M = 1 &,

!***************************************************************!
!*                                                             *!
!* Condition codes.                                            *!
!*                                                             *!
!***************************************************************!

  T  = 0  &, ! True
  F  = 1  &, ! False
  HI = 2  &, ! HIgh
  LS = 3  &, ! Low or Same
  CC = 4  &, ! Carry Clear
  CS = 5  &, ! Carry Set
  NE = 6  &, ! Not Equal
  EQ = 7  &, ! EQual
  VC = 8  &, ! oVerflow Clear
  VS = 9  &, ! oVerflow Set
  PL = 10 &, ! PLus
  MI = 11 &, ! MInus
  GE = 12 &, ! Greater or Equal
  LT = 13 &, ! Less Than
  GT = 14 &, ! Greater Than
  LE = 15 &, ! Less or Equal
```

```
!*****************************************************************!
!*                                                               *!
!* Addressing mode macros.                                       *!
!*                                                               *!
!*****************************************************************!

  sDR (rn) =
    if %rn eql '^SR$' then
       {
       Src_Mode = 7;
       Src_Rn = 4
       }
    else
       {
       Src_Mode = DR;
       Src_Rn = rn
       } &,

  sAR (rn) =
    Src_Mode = AR;
    Src_Rn = rn &,

  sIR (rn) =
    Src_Mode = IR;
    Src_Rn = rn &,

  sAI (rn) =
    Src_Mode = AI;
    Src_Rn = rn &,

  sAD (rn) =
    Src_Mode = AD;
    Src_Rn = rn &,

  dDR (rn) =
    Dst_Mode0 = DR ^ -2;
    Dst_Mode1 = DR;
    Dst_Rn = rn &,

  dAR (rn) =
    Dst_Mode0 = AR ^ -2;
    Dst_Mode1 = AR;
    Dst_Rn = rn &,

  dIR (rn) =
    Dst_Mode0 = IR ^ -2;
    Dst_Mode1 = IR;
    Dst_Rn = rn &,

  dAI (rn) =
    Dst_Mode0 = AI ^ -2;
    Dst_Mode1 = AI;
    Dst_Rn = rn &,
dAD (rn) =
  Dst_Mode0 = AD ^ -2;
  Dst_Mode1 = AD;
  Dst_Rn = rn &,

app_W (addr) =
  if length eql 2 then
    {
     I2 = addr ^ -8;
     I3 = addr
    };
  if length eql 4 then
    {
     I4 = addr ^ -8;
     I5 = addr
    };
  if length eql 6 then
    {
     I6 = addr ^ -8;
     I7 = addr
    };
  length = length + 2 &,

app_L (addr) =
  if length eql 2 then
    {
     I2 = addr ^ -24;
     I3 = addr ^ -16;
     I4 = addr ^ -8;
     I5 = addr
    };
  if length eql 4 then
    {
     I4 = addr ^ -24;
     I5 = addr ^ -16;
     I6 = addr ^ -8;
     I7 = addr
    };
  if length eql 6 then
    {
     I6 = addr ^ -24;
     I7 = addr ^ -16;
     I8 = addr ^ -8;
     I9 = addr
    };
  length = length + 4 &,

app_X (x_disp,x_ind,x_rn,x_size) =
  if length eql 2 then
    {
     T_Ind = x_ind;
     T_Rn = x_rn;
```

```
        T_Size = x_size - 1;
        T_Disp = x_disp
       };
     if length eql 4 then
       {
        F_Ind = x_ind;
        F_Rn = x_rn;
        F_Size = x_size - 1;
        F_Disp = x_disp
       };
     if length eql 6 then
       {
        S_Ind = x_ind;
        S_Rn = x_rn;
        S_Size = x_size - 1;
        S_Disp = x_disp
       };
     length = length + 2 &,

   sDS (rn,disp) =
     Src_Mode = DS;
     Src_Rn = rn;
     app_W (disp) &,

   sIX (rn,x_disp,x_ind,x_rn,x_size) =
     Src_Mode = IX;
     Src_Rn = rn;
     app_X (x_disp,x_ind,x_rn,x_size) &,

   sSH (addr) =
     Src_Mode = SP;
     Src_Rn = SH;
     app_W (addr) &,

   sLN (addr) =
     Src_Mode = SP;
     Src_Rn = LN;
     app_L (addr) &,

   sPD (disp) =
     Src_Mode = SP;
     Src_Rn = PD;
     app_W (disp) &,

   sPX (x_disp,x_ind,x_rn,x_size) =
     Src_Mode = SP;
     Src_Rn = PX;
     app_X (x_disp,x_ind,x_rn,x_size) &,

   dDS (rn,disp) =
     Dst_Mode0 = DS ^ -2;
     Dst_Mode1 = DS;
     Dst_Rn = rn;
```

```
      app_W (disp) &,

    dIX (rn,x_disp,x_ind,x_rn,x_size) =
      Dst_Mode0 = IX ^ -2;
      Dst_Mode1 = IX;
      Dst_Rn = rn;
      app_X (x_disp,x_ind,x_rn,x_size) &,

    dSH (addr) =
      Dst_Mode0 = SP ^ -2;
      Dst_Mode1 = SP;
      Dst_Rn = SH;
      app_W (addr) &,

    dLN (addr) =
      Dst_Mode0 = SP ^ -2;
      Dst_Mode1 = SP;
      Dst_Rn = LN;
      app_L (addr) &,

    dPD (disp) =
      Dst_Mode0 = SP ^ -2;
      Dst_Mode1 = SP;
      Dst_Rn = PD;
      app_W (disp) &,

    dPX (x_disp,x_ind,x_rn,x_size) =
      Dst_Mode0 = SP ^ -2;
      Dst_Mode1 = SP;
      Dst_Rn = PX;
      app_X (x_disp,x_ind,x_rn,x_size) &,

    adr_s (src) =
      if 1 then {s}src &,

    adr_d (dst) =
      if 1 then {d}dst &,

  !*****************************************************!
  !*                                                   *!
  !* instructions and their common macros in alpha-    *!
  !* betical order.                                    *!
  !*                                                   *!
  !*****************************************************!

    abcd (r_m,Dy_Ay,Dx_Ax) =
      Aux_Op = 1;
      R_M = r_m;
      Src_Rn = Dy_Ay;
      Dst_Rn = Dx_Ax &,

    ABCD (r_m,Dy_Ay,Dx_Ax) =
      Opcode = 12;
  abcd (r_m,Dy_Ay,Dx_Ax) $ &,

bIM (const) =
  I2 = 0;
  I3 = const;
  length = length + 2 &,

wIM (const) =
  I2 = const ^ -8;
  I3 = const;
  length = length + 2 &,

lIM (const) =
  I2 = const ^ -24;
  I3 = const ^ -16;
  I4 = const ^ -8;
  I5 = const;
  length = length + 4 &,

add (size,ea_Dn,Dn_ea) =
  Size = size;
  if 'Dn_ea eql "^D0$" or
     'Dn_ea eql "^D1$" or
     'Dn_ea eql "^D2$" or
     'Dn_ea eql "^D3$" or
     'Dn_ea eql "^D4$" or
     'Dn_ea eql "^D5$" or
     'Dn_ea eql "^D6$" or
     'Dn_ea eql "^D7$" then
    {
     Aux_Op = 0;
     Dst_Rn = Dn_ea;
     if 'ea_Dn eql "^IM.*" then
       {
        Src_Mode = SP;
        Src_Rn = IM;
        if 'size eql "^B$" then
          {
           if 1 then {b}ea_Dn
          };
        if 'size eql "^W$" then
          {
           if 1 then {w}ea_Dn
          };
        if 'size eql "^L$" then
          {
           if 1 then {l}ea_Dn
          }
       }
     else
       {
        adr_s (ea_Dn)
       }
```

```
        }
      else
        {
         Aux_Op = 1;
         Dst_Rn = ea_Dn;
         adr_s (Dn_ea)
        } &,

  ADD (size,ea_Dn,Dn_ea) =
    Opcode = 13;
    add (size,ea_Dn,Dn_ea) $ &,

  adda (size,ea,An) =
    Dst_Mode1 = 3;
    if 'size eql "^W$" then
        {
         Aux_Op = 0
        }
      else
        {
         Aux_Op = 1
        };
    Dst_Rn = An;
    adr_s (ea) &,

  ADDA (size,ea,An) =
    Opcode = 13;
    adda (size,ea,An) $ &,

  addi (size,data,ea) =
    Size = size;
    if 'size eql "^B$" then
        {
         I2 = 0;
         I3 = data;
         length = length + 2
        };
    if 'size eql "^W$" then
        {
         I2 = data ^ -8;
         I3 = data;
         length = length + 2
        };
    if 'size eql "^L$" then
        {
         I2 = data ^ -24;
         I3 = data ^ -16;
         I4 = data ^ -8;
         I5 = data;
         length = length + 4
        };
    adr_s (ea) &,
ADDI (size,data,ea) =
  IO = 6;
  addi (size,data,ea) $ &,

addq (size,data,ea) =
  Opcode = 5;
  Dst_Rn = data;
  Size = size;
  adr_s (ea) &,

ADDQ (size,data,ea) =
  Aux_Op = 0;
  addq (size,data,ea) $ &,

addx (size,r_m,Dy_Ay,Dx_Ax) =
  Size = size;
  abcd (r_m,Dy_Ay,Dx_Ax) &,

ADDX (size,r_m,Dy_Ay,Dx_Ax) =
  Opcode = 13;
  addx (size,r_m,Dy_Ay,Dx_Ax) $ &,

AND (size,ea_Dn,Dn_ea) =
  Opcode = 12;
  add (size,ea_Dn,Dn_ea) $ &,

ANDI (size,data,ea) =
  IO = 2;
  addi (size,data,ea) $ &,

asl (size,i_r,Dx_data,Dy) =
  Opcode = 14;
  Size = size;
  I_R = i_r;
  Dst_Rn = Dx_data;
  Src_Rn = Dy &,

ASL (size,i_r,Dx_data,Dy) =
  Aux_Op = 1;
  S_R = 0;
  R_M = 0;
  asl (size,i_r,Dx_data,Dy) $ &,

ASR (size,i_r,Dx_data,Dy) =
  Aux_Op = 0;
  S_R = 0;
  R_M = 0;
  asl (size,i_r,Dx_data,Dy) $ &,

aslm (ea) =
  Opcode = 14;
  Dst_Mode1 = 3;
  adr_s (ea) &,
ASLM (ea) =
  Dst_Rn = 0;
  Aux_Op = 1;
  aslm (ea) $ &,

ASRM (ea) =
  Dst_Rn = 0;
  Aux_Op = 0;
  aslm (ea) $ &,

BB (cc,label) =
  Opcode = 6;
  Condition = cc;
  I1 = label $ &,

BBRA (label) =
  Opcode = 6;
  Condition = T;
  I1 = label $ &,

BBSR (label) =
  Opcode = 6;
  Condition = F;
  I1 = label $ &,

bw (label) =
  Opcode = 6;
  I1 = 0;
  I2 = label ^ -8;
  I3 = label;
  length = length + 2 &,

BW (cc,label) =
  Condition = cc;
  bw (label) $ &,

BWRA (label) =
  Condition = T;
  bw (label) $ &,

BWSR (label) =
  Condition = F;
  bw (label) $ &,

bchg (s_d,Dn_data,ea) =
  Opcode = 0;
  if 's_d eql "^S$"
     {
      Dst_Rn = 4;
      Aux_Op = 0;
      I3 = Dn_data;
      length = length + 2
```

```
        }
      else
        {
         Dst_Rn = Dn_data;
         Aux_Op = 1;
        };
      adr_s (ea) &,

    BCHG (s_d,Dn_data,ea) =
      Dst_Model = 1;
      bchg (s_d,Dn_data,ea) $ &,

    BCLR (s_d,Dn_data,ea) =
      Dst_Model = 2;
      bchg (s_d,Dn_data,ea) $ &,

    BSET (s_d,Dn_data,ea) =
      Dst_Model = 3;
      bchg (s_d,Dn_data,ea) $ &,

    BTST (s_d,Dn_data,ea) =
      Dst_Model = 0;
      bchg (s_d,Dn_data,ea) $ &,

    chk (ea,Dn) =
      Opcode = 4;
      Dst_Rn = Dn;
      Aux_Op = 1;
      adr_s (ea) &,

    CHK (ea,Dn) =
      Dst_Model = 2;
      chk (ea,Dn) $ &,

    clr (size,ea) =
      Size = size;
      adr_s (ea) &,

    CLR (size,ea) =
      IO = 0x42;
      clr (size,ea) $ &,

    CMP (size,ea,Dn) =
      Opcode = 11;
      add (size,ea,Dn) $ &,

    CMPA (size,ea,An) =
      Opcode = 11;
      adda (size,ea,An) $ &,

    CMPI (size,data,ea) =
      IO = 12;
      addi (size,data,ea) $ &,
CMPM (size,Ay,Ax) =
  Opcode = 11;
  Aux_Op = 1;
  Src_Rn = 1;
  Size = size;
  Dst_Rn = Ax;
  Src_Rn = Ay $ &,

db (Dn,label) =
  Opcode = 5;
  Dst_Mode1 = 3;
  Src_Mode = 1;
  Src_Rn = Dn;
  I2 = label ^ -8;
  I3 = label;
  length = length+ 2 &,

DB (cc,Dn,label) =
  Condition = cc;
  db (Dn,label) $ &,

DBRA (Dn,label) =
  Condition = F;
  db (Dn,label) $ &,

divs (ea,Dn) =
  Opcode = 8;
  Dst_Rn = Dn;
  Dst_Mode1 = 3;
  adr_s (ea) $ &,

DIVS (ea,Dn) =
  Aux_Op = 1;
  divs (ea,Dn) $ &,

DIVU (ea,Dn) =
  Aux_Op = 0;
  divs (ea,Dn) $ &,

EOR (size,Dn,ea) =
  Opcode = 11;
  add (size,Dn,ea) $ &,

EORI (size,data,ea) =
  I0 = 10;
  addi (size,data,ea) $ &,

EXG (Dx_Ax,Dy_Ay) =
  Opcode = 12;
  Dst_Rn = Dx_Ax;
  Aux_Op = 1;
  Src_Rn = Dy_Ay;
  if 'Dx_Ax eql "^D*" and
     'Dy_Ay eql "^D*" then
    {
     Dst_Mode1 = 1;
     Src_Mode = 0
    };
  if 'Dx_Ax eql "^A*" and
     'Dy_Ay eql "^A*" then
    {
     Dst_Mode1 = 1;
     Src_Mode = 1
    }
  else
    {
     Dst_Mode1 = 2;
     Src_Mode = 1
    } $ &,

EXT (size,Dn) =
  Opcode = 4;
  Dst_Rn = 4;
  Aux_Op = 0;
  Size = size + 1;
  Src_Mode = 0;
  Src_Rn = Dn $ &,

jmp (ea) =
  IO = 0x4e;
  adr_s (ea) &,

JMP (ea) =
  Dst_Mode1 = 3;
  jmp (ea) $ &,

JSR (src) =
  Dst_Mode1 = 2;
  jmp (ea) $ &,

LEA (ea,An) =
  Dst_Mode1 = 3;
  chk (ea,An) $ &,

link (An) =
  IO = 0x4d;
  Dst_Mode1 = 1;
  Src_Rn = An &,

LINK (An,disp) =
  Src_Mode = 2;
  link (An);
  I2 = disp ^ -8;
  I3 = disp;
  length = length + 2 $ &,
LSL (size,i_r,Dx_data,Dy) =
  Aux_Op = 1;
  S_R = 0;
  R_M = 1;
  asl (size,i_r,Dx_data,Dy) $ &,

LSR (size,i_r,Dx_data,Dy) =
  Aux_Op = 0;
  S_R = 0;
  R_M = 1;
  asl (size,i_r,Dx_data,Dy) $ &,

LSLM (ea) =
  Dst_Rn = 1;
  Aux_Op = 1;
  aslm (ea) $ &,

LSRM (ea) =
  Dst_Rn = 1;
  Aux_Op = 0;
  aslm (ea) $ &,

MOVE (size,ea1,ea2) =
  if 'size eql "^B$" then
    {
     Opcode = 1
    };
  if 'size eql "^W$" then
    {
     Opcode = 3
    };
  if 'size eql "^L$" then
    {
     Opcode = 2
    };
  if 'ea1 eql "^IM.*" then
    {
     Src_Mode = SP;
     Src_Rn = IM;
     if 'size eql "^B$" then
       {
        if 1 then {b}ea1
       };
     if 'size eql "^W$" then
       {
        if 1 then {w}ea1
       };
     if 'size eql "^L$" then
       {
        if 1 then {l}ea1
       }
    }
  else
    {
     adr_s (ea1)
    };
  adr_d (ea2) $ &,

lccr (ea) =
  Opcode = 4;
  Dst_Mode0 = 0;
  Dst_Mode1 = 3;
  adr_s (ea) &,

! Move to CCR, LoaD CCR
LDCCR (ea) =
  Dst_Rn = 2;
  lccr (ea) $ &,

! Move to SR, LoaD SR
LDSR (ea) =
  Dst_Rn = 3;
  lccr (ea) $ &,

! Move from SR, STore SR
STSR (ea) =
  Dst_Rn = 0;
  lccr (ea) $ &,

! Move to USP, LoaD USP
LDUSP (An) =
  Src_Mode = 4;
  link (ea) $ &,

! Move from USP, STore USP
STUSP (ea) =
  Src_Mode = 5;
  link (ea) $ &,

MOVEA (size,ea,An) =
  if 'size eql "^W$" then
    {
     Opcode = 3
    }
  else
    {
     Opcode = 2
    };
  Dst_Rn = an;
  Dst_Mode0 = 0;
  Dst_Mode1 = 1;
  if 'ea eql "^IM.*" then
    {
     Src_Mode = SP;
     Src_Rn = IM;
```

```
        if 'size eql "^W$" then
          {
           if 1 then {w}ea
          }
        else
          {
           if 1 then {l}ea
          }
      }
    else
      {
       adr_s (ea)
      } $ &,

  MOVEQ (data,Dn) =
    Opcode = 7;
    Dst_Rn = Dn;
    Aux_Op = 0;
    I1 = data $ &,

  muls (ea,Dn) =
    Opcode = 12;
    Dst_Rn = Dn;
    Dst_Model = 3;
    adr_s (ea) &,

  MULS (ea,Dn) =
    Aux_Op = 1;
    muls (ea,Dn) $ &,

  MULU (ea,Dn) =
    Aux_Op = 0;
    muls (ea,Dn) $ &,

  NBCD (ea) =
    I0 = 0x48;
    Dst_Model = 0;
    adr_s (ea) $ &,

  NEG (size,ea) =
    I0 = 0x44;
    clr (size,ea) $ &,

  NEGX (size,ea) =
    I0 = 0x40;
    clr (size,ea) $ &,

  nop =
    I0 = 0x4e;
    Dst_Model = 1;
    Src_Mode = 6 &,

  NOP =
    Src_Rn = 1;
    nop $ &,

  NOT (size,ea) =
    IO = 0x46;
    clr (size,ea) $ &,

  OR (size,ea_Dn,Dn_ea) =
    Opcode = 8;
    add (size,ea_Dn,Dn_ea) $ &,

  ORI (size,data,ea) =
    IO = 0;
    addi (size,data,ea) $ &,

  PEA (src) =
    IO = 0x48;
    Dst_Mode = 1;
    adr_s (src) $ &,

  RESET =
    Src_Rn = 0;
    nop $ &,

  ROL (size,i_r,Dx_data,Dy) =
    Aux_Op = 1;
    S_R = 1;
    R_M = 1;
    asl (size,i_r,Dx_data,Dy) $ &,

  ROR (size,i_r,Dx_data,Dy) =
    Aux_Op = 0;
    S_R = 1;
    R_M = 1;
    asl (size,i_r,Dx_data,Dy) $ &,

  ROLM (ea) =
    Dst_Rn = 3;
    Aux_Op = 1;
    aslm (ea) $ &,

  RORM (ea) =
    Dst_Rn = 3;
    Aux_Op = 0;
    aslm (ea) $ &,

  ROXL (size,i_r,Dx_data,Dy) =
    Aux_Op = 1;
    S_R = 1;
    R_M = 0;
    asl (size,i_r,Dx_data,Dy) $ &,

  ROXR (size,i_r,Dx_data,Dy) =
  Aux_Op = 0;
  S_R = 1;
  R_M = 0;
  asl (size,i_r,Dx_data,Dy) $ &,

ROXLM (ea) =
  Dst_Rn = 2;
  Aux_Op = 1;
  aslm (ea) $ &,

ROXRM (ea) =
  Dst_Rn = 2;
  Aux_Op = 0;
  aslm (ea) $ &,

RTE =
  Src_Rn = 3;
  nop $ &,

RTR =
  Src_Rn = 7;
  nop $ &,

RTS =
  Src_Rn = 5;
  nop $ &,

SBCD (r_m,Dy_Ay,Dx_Ax) =
  Opcode = 8;
  abcd (r_m,Dy_Ay,Dx_Ax) $ &,

S (cc,ea) =
  Opcode = 5;
  Condition = cc;
  Dst_Mode1 = 3;
  adr_s (ea) $ &,

STOP (data) =
  Src_Rn = 2;
  nop;
  I2 = data ^ -8;
  I3 = data;
  length = length + 2 $ &,

SUB (size,ea_Dn,Dn_ea) =
  Opcode = 9;
  add (size,ea_Dn,Dn_ea) $ &,

SUBA (size,ea,An) =
  Opcode = 9;
  adda (size,ea,An) $ &,

SUBI (size,data,ea) =
    IO = 8;
    addi (size,data,ea) $ &,

  SUBQ (size,data,ea) =
    Aux_Op = 1;
    addq (size,data,ea) $ &,

  SUBX (size,r_m,Dy_Ay,Dx_Ax) =
    Opcode = 9;
    addx (size,r_m,Dy_Ay,Dx_Ax) $ &,

  SWAP (Dn) =
    IO = 0x48;
    Dst_Model = 1;
    Src_Mode = 0;
    Src_Rn = Dn $ &,

  TAS (ea) =
    IO = 0x4a;
    Dst_Model = 3;
    adr_s (ea) $ &,

  TRAP (vector) =
    IO = 0x4d;
    Dst_Model = 1;
    I_R = 0;
    S_R = 0;
    R_M = vector ^ -3;
    Src_Rn = vector $ &,

  TRAPV =
    Src_Rn = 6;
    nop $ &,

  TST (size,ea) =
    IO = 0x4a;
    clr (size,ea) $ &,

  UNLK (An) =
    Src_Mode = 3;
    link (An) $ &,

!****************************************************!
!*                                                  *!
!* Psuedo instructions for constants.               *!
!*                                                  *!
!****************************************************!

  CB (const) =
    IO = const;
    length = 1 $ &,

  CW (const) =
    I0 = const ^ -8;
    I1 = const $ 8,

  CL (const) =
    I0 = const ^ -24;
    I1 = const ^ -16;
    I2 = const ^ -8;
    I3 = const;
    length = 4 $ &$
```

Appendix E: MC68000 Linking/Loader Description

```
!*************************************************!
!*                                               *!
!* m68000.i.                                     *!
!* Linking Loader description for Motorola 68000 *!
!* microprocessor.                               *!
!* No input requirements.                        *!
!* Generates m68000.a on compilation.            *!
!* Use "inter" to compile.                       *!
!*                                               *!
!* Author   : Samir S. Shah.                     *!
!* Date     : Fall 1979.                         *!
!* Modified : Samir S. Shah.                     *!
!* Date     : Spring 1981.                       *!
!*                                               *!
!*************************************************!

!*************************************************!
!*                                               *!
!* Declaration for instruction lengths and widths. *!
!* Memory is byte addressable,default instruction *!
!* length is 2 bytes,maximum instruction length is *!
!* 10 bytes.                                     *!
!*                                               *!
!*************************************************!

instr
  I[10,2]<8>$


format

!*************************************************!
!*                                               *!
!* Fields of instruction bytes zero and one.     *!
!*                                               *!
!*************************************************!

  Opcode    = I[0]<7:4>,
  Condition = I[0]<3:0>,
  Dst_Rn    = I[0]<3:1>,
  Dst_Mode0 = I[0]<0>,
  Aux_Op    = I[0]<0>,
  Dst_Mode1 = I[1]<7:6>,
  Size      = I[1]<7:6>,
  Src_Mode  = I[1]<5:3>,
  I_R       = I[1]<5>,
  S_R       = I[1]<4>,
  R_M         = I[1]<3>,
  Src_Rn      = I[1]<2:0>,

!*****************************************************!
!*                                                   *!
!* Fields of instruction bytes Two and three.        *!
!*                                                   *!
!*****************************************************!

  T_Ind  = I[2]<7>,
  T_Rn   = I[2]<6:4>,
  T_Size = I[2]<3>,
  T_Disp = I[3]<7:0>,

!*****************************************************!
!*                                                   *!
!* Fields of instruction bytes Four and five.        *!
!*                                                   *!
!*****************************************************!

  F_Ind  = I[4]<7>,
  F_Rn   = I[4]<6:4>,
  F_Size = I[4]<3>,
  F_Disp = I[5]<7:0>,

!*****************************************************!
!*                                                   *!
!* Fields of instruction bytes Six and seven.        *!
!*                                                   *!
!*****************************************************!

  S_Ind  = I[6]<7>,
  S_Rn   = I[6]<6:4>,
  S_Size = I[6]<3>,
  S_Disp = I[7]<7:0>,

!*****************************************************!
!*                                                   *!
!* Shorter names for instruction bytes.              *!
!*                                                   *!
!*****************************************************!

  I0 = I[0]<7:0>,
  I1 = I[1]<7:0>,
  I2 = I[2]<7:0>,
  I3 = I[3]<7:0>,
  I4 = I[4]<7:0>,
  I5 = I[5]<7:0>,
  I6 = I[6]<7:0>,
  I7 = I[7]<7:0>,
  I8 = I[8]<7:0>,
  I9 = I[9]<7:0>$
```

```
space
  <0:32767>$

transfer
  {new
   I0 = 0x4e $
   I1 = 0xf9 $
   I2 = address ^ -24 $
   I3 = address ^ -16 $
   I4 = address ^ -8 $
   I5 = address $
  }

mode
  case (labelcnt eql 2) and
       (length eql 10) :
    I2 = address[1] ^ -24 $
    I3 = address[1] ^ -16 $
    I4 = address[1] ^ -8 $
    I5 = address[1] $
    I6 = address[5] ^ -24 $
    I7 = address[5] ^ -16 $
    I8 = address[5] ^ -8 $
    I9 = address[5] $
  break$
  esac,
  case Opcode eql 6:
       I1 = address - , - 2$
       break$
  esac,
  default:
  esac$
```

Appendix F: Simulation Test Routines

This appendix identifies the Metamicro test routines processed for each of the MC68000 instruction and exception models that were simulated. They are:

21. Address Error Exception F-8

1. MOVE.W D1,D2

```
include mc68000.m$
begin
  MOVE (W,DR(D1),DR(D2))      ! Move contents of data register
  MOVE (W,DR(D1),DR(D2))      ! D1 to data register D2
  MOVE (W,DR(D1),DR(D2))
  MOVE (W,DR(D1),DR(D2))
  MOVE (W,DR(D1),DR(D2))
  MOVE (W,DR(D1),DR(D2))
  JMP  (IR(A0))               ! Jump to address pointed to by
                              ! address register A0
end
```

2. MOVE.W D1,(A1)

```
include mc68000.m$
begin
  MOVE (W,DR(D1),IR(A1))      ! Move contents of data register
  MOVE (W,DR(D1),IR(A1))      ! D1 to memory location pointed
  MOVE (W,DR(D1),IR(A1))      ! to by address register A1
  MOVE (W,DR(D1),IR(A1))
  MOVE (W,DR(D1),IR(A1))
  MOVE (W,DR(D1),IR(A1))
  JMP  (IR(A0))               ! Jump to address pointed to by
                              ! address register A0
end
```

3. MOVE.L D1,A1

```
include mc68000.m$
begin
    . = 0x1000$               ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))      ! Set supervisor mode bit to zero
                              ! (user mode)
  MOVE (L,DR(D1),AR(A1))      ! Move 32-bit contents of data
  MOVE (L,DR(D1),AR(A1))      ! register D1 to address register
                              ! A1
  JMP  (IR(A0))               ! Jump to address pointed to by
                              ! address register A0
end
```

4. MOVE.W D1,(A1)+

```
include mc68000.m$
begin
    . = 0x1000$               ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))      ! Set supervisor mode bit to zero
                              ! (user mode)
  MOVE (W,DR(D1),AI(A1))      ! Move contents of data
  MOVE (W,DR(D1),AI(A1))     ! register D1 to memory location
                             ! pointed to by address register A1
                             ! and then increment A1 by 2
  MOVE (L,DR(D2),AR(A1))     ! Re-initialize address register A1
  JMP  (IR(A0))              ! Jump to address pointed to by
                             ! address register A0
end
```

5. MOVE.W D1,04(A1)

```
include mc68000.m$
begin
    . = 0x1000$              ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),DS(A1,4))   ! Add displacement (4) to address
                             ! register A1 to form address
                             ! which will receive the contents
                             ! of data register D1
  MOVE (W,DR(D1),DS(A1,8))   ! Repeat with displacement 8
  JMP  (IR(A0))              ! Jump to address pointed to by
                             ! address register A0
end
```

6. MOVE.W D1,04(A1,D7)

```
include mc68000.m$
begin
    . = 0x1000$              ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),IX(A1,4,0,D7,1)) ! Sum displacement (4), the
                             ! contents of data register D7 and
                             ! address register A1 to form ad-
                             ! dress of memory location to
                             ! receive data register D1
  MOVE (W,DR(D1),IX(A1,4,0,D7,1)) ! Repeat with displacement 8
  JMP  (IR(A0))              ! Jump to address pointed to by
                             ! address register A0
end
```

7. MOVE.W D1,$2004

```
include mc68000.m$
begin
    . = 0x1000$              ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),SH(0x2004))! Move contents of data register
                             ! D1 to memory location 2004 hex
  MOVE (W,DR(D1),SH(0x2008))! Repeat at location 2008 hex
  JMP  (IR(A0))              ! Jump to address pointed to by
```

```
                                      ! address register A0
end
```

8. MOVE.W A1,D3

```
include mc68000.m$
begin
     . = 0x1000$                      ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))              ! Set supervisor mode bit to zero
                                      ! (user mode)
  MOVE (W,AR(A1),DR(D3))              ! Move contents of address register
  MOVE (W,AR(A1),DR(D3))              ! A1 to data register D3
  JMP  (IR(A0))                       ! Jump to address pointed to by
                                      ! address register A0
end
```

9. MOVE.W (A1),D2

```
include mc68000.m$
begin
     . = 0x1000$                      ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))              ! Set supervisor mode bit to zero
                                      ! (user mode)
  MOVE (W,IR(A1),DR(D2))              ! Move contents of memory location
                                      ! pointed to by address register
  MOVE (W,IR(A1),DR(D2))              ! A1 to data register D2
  JMP  (IR(A0))                       ! Jump to address pointed to by
                                      ! address register A0
end
```

10. MOVE.W (A1)+,D6

```
include mc68000.m$
begin
     . = 0x1000$                      ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))              ! Set supervisor mode bit to zero
                                      ! (user mode)
  MOVE (W,AI(A1),DR(D6))              ! Move contents of memory location
                                      ! pointed to by address register
  MOVE (W,AI(A1),DR(D6))              ! A1 to data register D6
                                      ! then increment A1 by 2
  JMP  (IR(A0))                       ! Jump to address pointed to by
                                      ! address register A0
end
```

11. MOVE.W -(A1),D4

```
include mc68000.m$
begin
     . = 0x1000$                      ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))              ! Set supervisor mode bit to zero
```

```
                                     ! (user mode)
    MOVE (W,AD(A1),DR(D4))           ! Decrement address register A1
                                     ! by 2 and then use to identify
                                     ! memory location to receive con-
                                     ! tents of data register D4
    MOVE (W,AD(A1),DR(D3))           ! Repeat for data register D3
    MOVE (L,DR(D2),AR(A1))           ! Reset A1
    JMP  (IR(A0))                    ! Jump to address pointed to by
                                     ! address register A0
end
```

12. MOVE.W 04(A1),D1

```
include mc68000.m$
begin
     . = 0x1000$                     ! Load routine at address 1000 hex
    ANDI (W,0xdfff,DR(SR))           ! Set supervisor mode bit to zero
                                     ! (user mode)
    MOVE (W,DS(A1,4),DR(D1))         ! Move contents of memory location
                                     ! determined by summing displace-
                                     ! ment 4 and contents of address
                                     ! register A1-to data register D1
    MOVE (W,DS(A1,8),DR(D2))         ! Repeat with displacement 8 and
                                     ! data register D2
    JMP  (IR(A0))                    ! Jump to address pointed to by
                                     ! address register A0
end
```

13. MOVE.W 04(A1,D7),D2

```
include mc68000.m$
begin
     . = 0x1000$                     ! Load routine at address 1000 hex
    ANDI (W,0xdfff,DR(SR))           ! Set supervisor mode bit to zero
                                     ! (user mode)
    MOVE (W,IX(A1,4,0,D7,1),DR(D2)) ! Move contents of memory
                                     ! location determined by the sum of
                                     ! displacement 4, contents of data
                                     ! register D7, and address register
                                     ! A1 - to data register D2
    MOVE (W,IX(A1,4,0,D7,1),DR(D3)) ! Repeat for data register D3
    JMP  (IR(A0))                    ! Jump to address pointed to by
                                     ! address register A0
end
```

14. MOVE.W $2004,D5

```
include mc68000.m$
begin
     . = 0x1000$                     ! Load routine at address 1000 hex
    ANDI (W,0xdfff,DR(SR))           ! Set supervisor mode bit to zero
                                     ! (user mode)
```

```
  MOVE  (W,SH(0x2004),DR(D5)) ! Move word at memory location
                             ! 2004 hex into data register D5
  MOVE  (W,SH(0x2004),DR(D6)) ! Repeat for data register D6
  JMP   (IR(A0))             ! Jump to address pointed to by
                             ! address register A0
end
```

15. MOVE.W $2004,$2008

```
include mc68000.m$
begin
     . = 0x1000$             ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,LN(0x2004),LN(0x2008)) ! Move word at memory location
                             ! 2004 hex into memory location
                             ! 2008 hex
  JMP   (IR(A0))             ! Jump to address pointed to by
                             ! address register A0
end
```

16. MOVE.W #$5555,D1

```
include mc68000.m$
begin
     . = 0x1000$             ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,IM(0x5555),DR(D1)) ! Move data word 5555 hex into
  MOVE (W,IM(0x5555),DR(D1)) ! data register D1
  JMP   (IR(A0))             ! Jump to address pointed to by
                             ! address register A0
end
```

17. ADD.W D3,D5

```
include mc68000.m$
begin
     . = 0x1000$             ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))     ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),DR(D5))     ! Move contents of data register D1
                             ! into data register D5
  ADD   (W,DR(D3),D5)        ! Sum contents of data registers D3
                             ! and D5 and store in D5
  MOVE (W,DR(D5),IR(A2))     ! Move result to memory location
                             ! identified by address register A2
  JMP   (IR(A0))             ! Jump to address pointed to by
                             ! address register A0
end
```

18. BEQ START

```
include mc68000.m$
begin
  START:  MOVE  (W,DR(D1),DR(D3)) ! This move will clear Zero
                                  ! condition code of SR
          BB    (EQ,START)        ! Branch not taken
          MOVE  (W,DR(D2),DR(D3)) ! Set Zero condition code
          BB    (EQ,START)        ! Branch taken
          JMP   (IR(A0))          ! Jump to address pointed to
                                  ! by address register A0
end
```

19. BTST D1,(A1)

```
include mc68000.m$
begin
  MOVE (W,DR(D2),DR(D3)) ! Fill prefetch queue
  BTST (R,D1,IR(A1))     ! Test bit identified by contents
                         ! of data register D1 in memory
                         ! location identified by address
                         ! register A1
  MOVE (W,DR(D2),DR(D3)) ! To determine completion of BTST
  JMP  (IR(A0))            ! Jump to address pointed to by
                           ! address register A0
end
```

20. Illegal Instruction Exception

```
include mc68000.m$
begin
     . = 0x1000$            ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))    ! Set supervisor mode bit to zero
                            ! (user mode)
  MOVE (W,DR(D1),DR(D2))    ! Surround illegal instruction with
  MOVE (W,DR(D1),DR(D2))    ! MOVE's to fill prefetch queue
  MOVE (W,DR(D1),DR(D2))    ! and isolate illegal instruction
  CW   (0x4afc)             ! Illegal Instruction
  MOVE (W,DR(D1),DR(D2))
  JMP  (IR(A0))             ! Jump to address pointed to by
                            ! address register A0
end
```

21. Address Error Exception

```
include mc68000.m$
begin
     . = 0x1000$            ! Load routine at address 1000 hex
  ANDI (W,0xdfff,DR(SR))    ! Set supervisor mode bit to zero
                            ! (user mode)
  MOVE (W,DR(D1),IR(A1))    ! Attempt to move word from data
                            ! register D1 to odd address con-
```

```
                                        ! tained in address register A1
    NOP                                 ! This instruction will be pre-
                                        ! fetched during address error
                                        ! instruction
    JMP  (IR(A0))                       ! Jump to address pointed to by
                                        ! address register A0
       . = 0x2040$                      ! Load exception handler routine
                                        ! at memory location 2040 hex
    ADDQ (L,4,AR(A7))                   ! Pop system stack to retrieve
    ADDQ (L,4,AR(A7))                   ! program counter pointing to
                                        ! JMP (A0) instruction
    RTE                                 ! Return from Exception
  end
```

Appendix G: Simulation Control Files

This appendix contains the global files used to build each simulation and output the data of interest.

Topology File

The following is an overlay of the topology files used to identify and bind each of the simulation's components for processing by N.mPc's Ecologist:

```
signal         DBUS(16),              ! Data Bus
               ABUS(23);              ! Address Bus

processor       cpu = "root.sim"      ! The file "root.sim"
                                      ! contains the compiled
                                      ! MC68000 model

time delay      1000000 ns;           ! A simulation delay
                                      ! corresponds to 1 ms

connections    DBUS = DBUS,           ! Connect processor's
                                      ! Data Bus pins to
                                      ! Data Bus
               ABUS = ABUS;           ! Ditto Address Bus Pins

initial           M = ROOT;           ! Load memory "M" with
                                      ! executable code contained
                                      ! in the file "ROOT"
```

This file will support the modeling of any of the instructions if the file "root.sim" is replaced with the compiled version of the instruction's hardware component in the "processor" declaration, and the name of the file containing the executable code for the instruction's test routine is substituted for the file "ROOT" in the "initial" declaration.

Simulate

The file "simulate" governed each simulation by determining when simulation breakpoints would occur and then executing the appropriate signal output file to display the desired data. It appears as follows:

```
setproc cpu                     ! Default processor
repeat bkpt :PHI1 eql 1         ! Break each time PHI1 is high
repeat trigger "signals1" 1     ! and execute file "signals1"
repeat bkpt :PHI2 eql 1         ! Break each time PHI2 is high
repeat trigger "signals2" 2     ! and execute file "signals2"
bkpt :T eql 0 after :T eql 9 after :IR eql 0h4ed0  ! Stop
                                ! simulation at instruction
                                ! following JMP (A0)
run                             ! Start simulation
```

Signals1/Signals2

Signals1 and Signals2 were executed at the appropriate simulation breakpoints to accomplish the actual data display. These two files differ only in the clock phase signal displayed. Signals1 will display the state of PHI1 as a part of its output while Signals2 displays PHI2. They both consist of the following Runtime statements:

```
base 10            ! Want clock cycle displayed in base 10
examine :T         ! Display clock cycle
examine :PHI1      ! Display phase 1 of current clock cycle
                   ! Signals2 will display phase 2
base 2             ! Display remaining data in base 2
examine :FC        ! Display Function Code Signals
examine :DTACKN    ! Display Data Transfer Acknowledge
examine :RW        ! Display Read/Write
examine :LDSN      ! Display Lower Data Strobe
examine :UDSN      ! Display Upper Data Strobe
examine :ASN       ! Display Address Strobe
examine :DBUS      ! Display state of Data Bus
```